Available online at www.sciencedirect.com

ScienceDirect

Journal of Computer and System Sciences 74 (2008) 671–696

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

On the Approximation of Correlation Clustering and Consensus Clustering ☆

Paola Bonizzoni [a], Gianluca Della Vedova [b,*], Riccardo Dondi [c], Tao Jiang [d,e]

[a] *Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Milano, Italy*
[b] *Dipartimento di Statistica, Università degli Studi di Milano-Bicocca, Milano, Italy*
[c] *Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali, Università degli Studi di Bergamo, Bergamo, Italy*
[d] *University of California, Riverside, CA, USA*
[e] *Department of Computer Science and Engineering, Tsinghua University, Beijing, China*

Received 17 March 2007; received in revised form 8 June 2007

Available online 16 June 2007

Abstract

The Correlation Clustering problem has been introduced recently [N. Bansal, A. Blum, S. Chawla, Correlation Clustering, in: Proc. 43rd Symp. Foundations of Computer Science, FOCS, 2002, pp. 238–247] as a model for clustering data when a binary relationship between data points is known. More precisely, for each pair of points we have two scores measuring the similarity and dissimilarity respectively, of the two points, and we would like to compute an optimal partition where the value of a partition is obtained by summing up the similarity scores of pairs involving points from the same cluster and the dissimilarity scores of pairs involving points from different clusters. A closely related problem is Consensus Clustering, where we are given a set of partitions and we would like to obtain a partition that best summarizes the input partitions. The latter problem is a restricted case of Correlation Clustering. In this paper we prove that Minimum Consensus Clustering is APX-hard even for three input partitions, answering an open question in the literature, while Maximum Consensus Clustering admits a PTAS. We exhibit a combinatorial and practical $\frac{4}{5}$-approximation algorithm based on a greedy technique for Maximum Consensus Clustering on three partitions. Moreover, we prove that a PTAS exists for Maximum Correlation Clustering when the maximum ratio between two scores is at most a constant.

Keywords: Correlation Clustering; Consensus Clustering; Approximation; APX-hardness; PTAS

1. Introduction

The problem of analyzing a set of points in order to isolate subsets of points that are closely related is known as *clustering*. Clustering is an important problem in computer science due to its broad applications in areas such as datamining, machine learning, and bioinformatics. An example application taken from [6] is to cluster a set of

☆ Research supported in part by NSF grant CCR-0309902, NSFC grant 60528001, National Key Project for Basic Research (973) grant 2002CB512801, and a fellowship from the Center for Advanced Study, Tsinghua University.
* Corresponding author.
E-mail address: gianluca.dellavedova@unimib.it (G. Della Vedova).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2007.06.024

documents into topics without having a clear and unambiguous notion of topic, but with a measure of similarity (or dissimilarity) between pairs of documents. We construct a graph G whose vertices are the documents. Each edge e is labeled with two weights, where the first weight $a(e)$ measures the similarity between the documents incident on e and the second weight $b(e)$ measures the distance or dissimilarity between the documents incident on e. Given a partition π of the vertex set of G, the value of π can be formally defined as the summation of $a(e)$ for each edge internal to a cluster and $b(e)$ for each edge whose endpoints are in two different clusters. The MAXIMUM CORRELATION CLUSTERING problem asks for a partition π of vertices of G of maximum value. An interesting property of this approach is that the number of clusters to be obtained is not predetermined *a priori*, as in the case of the k-median or k-center problems [16], and it depends uniquely on the instance.

There are in fact several versions of the problem. For instance, both the maximization and minimization versions of the problem have been studied in the literature and shown to have different approximation properties, as we will see in this paper.

The minimization version, called MINIMUM CORRELATION CLUSTERING, asks for a partition π of vertices of G whose value is defined by summing the weight $a(e)$ for each edge e whose endpoints are in two different clusters and weight $b(e)$ for each edge internal to a cluster. Moreover, if $a(e) = b(e) = 0$ is allowed for an edge e, then the underlying graph of the problem becomes a general graph instead of a complete graph. It is known that the minimization version of the problem, over general graphs has an $O(\log n)$-approximation algorithm [7–9], while MAXIMUM CORRELATION CLUSTERING over general graphs has a 0.7666-approximation algorithm [7,18].[1]

Unweighted versions of CORRELATION CLUSTERING have also been considered before. In this case all scores $a(e)$ are either 0 or 1 and $b(e) = 1 - a(e)$. The unweighted MINIMIZATION CORRELATION CLUSTERING is known to be APX-hard [6] while admitting a 4-approximation algorithm [7] and a randomized 3-approximation algorithm [1]. The unweighted MAXIMUM CORRELATION CLUSTERING admits a probabilistic PTAS [6]. An interesting restriction of the CORRELATION CLUSTERING is the case where the scores $b(e)$ satisfy the triangle inequality and $a(e) = 1 - b(e)$. These kinds of instances is particularly interesting, since they can be obtained for example by reducing Consensus Clustering to Correlation Clustering. In [13] a combinatorial algorithm of factor 3 for the minimization version of this restriction is presented, which was improved in [1], where an approximation algorithm of factor 2 is presented.

In this paper, we will describe a PTAS for MAXIMUM CORRELATION CLUSTERING on input graphs where the ratio between the largest and smallest weights is bounded by a constant. Such a restriction clearly implies that all weights are strictly larger than zero, and hence the result is applicable only to complete graphs. The PTAS is based on the smooth polynomial programming technique introduced in [3] and exploits the natural denseness of the problem.

Another variant of CORRELATION CLUSTERING is the CONSENSUS CLUSTERING problem. Here, we are given a set of partitions and we want to compute the *median* partition, i.e. a partition having the total maximum similarity (or minimum distance) to the input partitions. More precisely, in MINIMUM CONSENSUS CLUSTERING the distance between two partitions π_1 and π_2 is defined as the number of pairs which are clustered differently in π_1 and in π_2, that is the number of pairs co-clustered in π_1 and not co-clustered in π_2 plus the number of pairs co-clustered in π_2 and not co-clustered in π_1. The similarity between two partitions π_1 and π_2 is defined as the number of pairs co-clustered in both π_1 and π_2 plus the number of pairs not co-clustered in both π_1 and π_2. CONSENSUS CLUSTERING has been studied extensively in the literature [17,19], and its NP-hardness over general graphs is well known [17,19]. Recently more attention has been given to the problem because of its application in bioinformatics, in particular microarray data analysis. It is observed in [10,11] that microarray experiments provide measures of gene expression levels, and clustering genes with similar expression levels could provide information useful for the construction of genetic networks. Since different experimental conditions may result in significantly different expression data (thus partitions of genes), it is often useful to compute the consensus of the partitions given by a collection of gene expression data. It is easy to see that CONSENSUS CLUSTERING is actually a special case of (weighted) CORRELATION CLUSTERING. Consider an instance of CONSENSUS CLUSTERING. For each pair of elements of the universe, x_1 and x_2, define an edge $e = (x_1, x_2)$ with weight $a(e)$ equal to the number of input partitions containing x_1 and x_2 in the same set (i.e. they are *co-clustered*) and weight $b(e)$ equal to the number of input partitions containing x_1 and x_2 in different sets (i.e. they are *not co-clustered*). Then, solving CONSENSUS CLUSTERING on the instance would be equivalent to solving CORRELATION CLUSTERING. MINIMUM CONSENSUS CLUSTERING admits a $\frac{11}{7}$-approximation algorithm [1].

[1] The works of [7,18] were done independently; [7] proved a 0.7664 approximation ratio, which was improved to 0.7666 in [18].

Moreover, a number of heuristics have been proposed for CONSENSUS CLUSTERING, which are based on cutting-plane [14] and simulated annealing [10] techniques. In the latter paper, it was observed that the problem is trivially solvable for instances of at most two partitions, while an open question, as recently recalled in [1], is the complexity of the problem (minimization and maximization versions) for k input partitions, for any constant $k > 2$. In this paper, we settle the open question by showing that MINIMUM CONSENSUS CLUSTERING is APX-hard even on instances with three input partitions.

The paper is organized as follows. Some definitions required in the constructions and proofs are given in next section. In Section 3, we show that MINIMUM CONSENSUS CLUSTERING is APX-hard on instances with three partitions. In contrast, in Section 4, we show that the maximization version of CONSENSUS CLUSTERING, i.e. MAXIMUM CONSENSUS CLUSTERING, admits a PTAS. This is achieved by first showing that MAXIMUM CORRELATION CLUSTERING on instances with bounded weights has a PTAS, by using the smooth polynomial programming technique. In Section 5, we exhibit a combinatorial and practical $\frac{4}{5}$-approximation algorithm based on a greedy technique for MAXIMUM CONSENSUS CLUSTERING on three partitions. To our knowledge, this is the first non-trivial combinatorial approximation algorithm for CONSENSUS CLUSTERING. Moreover, the approximation factor of 0.8 achieved in our algorithm improves on the approximation factor obtained by applying the best algorithm for CORRELATION CLUSTERING based on semidefinite programming [7,18].

2. Preliminaries

In this section, we introduce some basic notations and definitions that we will need later. Let π be a partition and r_π be the characteristic vector associated with π defined as follows:

$$r_\pi(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ are co-clustered in } \pi, \\ 0 & \text{if } (i,j) \text{ are not co-clustered in } \pi. \end{cases}$$

A *correlation graph* is a weighted complete graph G such that each edge (u, v) is labeled with two weights $a(u, v)$ and $b(u, v)$, where $a(u, v)$ is the similarity between u and v, while $b(u, v)$ is the distance between u and v. We study the following problems on correlation graphs.

Minimum Correlation Clustering. We are given a correlation graph $G = (V, E)$ and we want to find a partition π of V minimizing the sum $\sum_{e=(i,j)}(r_\pi(i,j)b(i,j) + (1 - r_\pi(i,j))a(i,j))$.

Maximum Correlation Clustering. We are given a correlation graph $G = (V, E)$ and we want to find a partition π of V maximizing the sum $\sum_{e=(i,j)}(r_\pi(i,j)a(i,j) + (1 - r_\pi(i,j))b(i,j))$.

The notions of distance and similarity that we will employ for defining the problems of CONSENSUS CLUSTERING are the following:

Definition 2.1. Let U be a universe set and let π_1, π_2 be two partitions of U. Let $d(\pi_1, \pi_2)$ denote the *symmetric difference distance* defined as the number of pairs of elements co-clustered in exactly one partition in $\{\pi_1, \pi_2\}$. Observe that the symmetric difference distance between two partitions π_1 and π_2 corresponds to the l_1 distance between the characteristic vectors r_{π_1} and r_{π_2} associated with π_1 and π_2 respectively. Let $s(\pi_1, \pi_2)$ denote the *similarity measure* defined as the number of pairs of elements co-clustered in both partitions plus the number of pairs of elements not co-clustered in both partitions π_1 and π_2. Observe that $s(\pi_1, \pi_2) + d(\pi_1, \pi_2) = \binom{|U|}{2}$ and hence $s(\pi_1, \pi_2) = \binom{|U|}{2} - d(\pi_1, \pi_2)$.

Also, given two elements i, j of the universe set U and a set $\Pi = \{\pi_1, \ldots, \pi_k\}$ of partitions of U, we denote by $s_\Pi(i, j)$ (or simply $s(i, j)$ whenever Π is known from the context) and $d_\Pi(i, j)$ (or simply $d(i, j)$) respectively, the number of partitions of Π in which i, j are co-clustered and are not co-clustered. Clearly, for each pair (i, j), $d_\Pi(i, j) + s_\Pi(i, j) = k$.

Minimum Consensus Clustering. We are given a set $\{\pi_1, \pi_2, \ldots, \pi_k\}$ of partitions over universe U and we want to find a partition π of the elements of U minimizing $\sum_{i=1}^{k} d(\pi_i, \pi)$.

Maximum Consensus Clustering. We are given a set $\{\pi_1, \pi_2, \ldots, \pi_k\}$ of partitions over universe U and we want to find a partition π of the elements of U maximizing $\sum_{i=1}^{k} s(\pi_i, \pi)$, called the *similarity cost*.

Observe that the *cost* $c(\pi)$ of a solution π of MINIMUM CONSENSUS CLUSTERING over an instance Π can also be expressed as:

$$\sum_{i<j} \big(r_\pi(i,j)d_\Pi(i,j) + \big(1 - r_\pi(i,j)\big)s_\Pi(i,j)\big). \tag{1}$$

Analogously, the *similarity value* $v(\pi)$ of a solution π of MAXIMUM CONSENSUS CLUSTERING over an instance Π can be defined as:

$$\sum_{i<j} \big(r_\pi(i,j)s_\Pi(i,j) + \big(1 - r_\pi(i,j)\big)d_\Pi(i,j)\big). \tag{2}$$

Comparing expressions 1 and 2 with the definition of MINIMUM CORRELATION CLUSTERING and MAXIMUM CORRELATION CLUSTERING, it is easy to see that CONSENSUS CLUSTERING corresponds to CORRELATION CLUSTERING with $a(i,j) = s_\Pi(i,j)$ and $b(i,j) = d_\Pi(i,j)$.

In what follows we show some properties of a restricted instance of CONSENSUS CLUSTERING consisting exactly of three partitions: we call this case 3-CONSENSUS CLUSTERING, in short 3CC. We will consider also the restricted case in which no pair of elements is co-clustered in all three partitions: we call this case MINIMUM RESTRICTED 3-CONSENSUS CLUSTERING, in short MR3CC. A fundamental notion used in the paper and in our reduction is that of 2-component of an instance Π of 3CC over universe U.

Definition 2.2. Let Π be an instance of 3CC over universe U. A subset X of U such that $|X| \geqslant 2$ is a 2-*component* of Π if each pair of elements of X is co-clustered in at least two input partitions of Π and X is a maximal subset of U with such a property.

It is possible to compute efficiently the 2-components of an instance Π of 3CC as shown by the following proposition.

Lemma 2.1. *Let Π be an instance of* 3-CONSENSUS CLUSTERING, *then a subset $X \subseteq U$, with $|X| \geqslant 2$ is a subset of a 2-component of Π if and only if there exist two sets A, B, with $A \in \pi_i$ and $B \in \pi_j$, $i \neq j$, $\pi_i, \pi_j \in \Pi$, and such that $X \subseteq A \cap B$.*

Proof. The *if* direction is a trivial consequence of the definition of 2-component, hence we can concentrate on the *only if* part. First notice that being a subset of a 2-component is a hereditary property, hence if X is not a subset of a 2-component then no superset of X can be a subset of a 2-component.

The proof of the proposition is by induction on $n = |X|$; clearly if $n = 2$ the statement holds. Now assume that the proposition holds for any subset with up to $n-1$ elements and let X be the set $\{x_1, x_2, \ldots, x_n\}$, with $n \geqslant 3$. By inductive hypothesis any subset X_1 of X is a subset of a 2-component if and only if X_1 is contained in two sets A, B with $A \in \pi_i$, $B \in \pi_j$.

Assume that X is a subset of a 2-component. Then, by induction each of the two subsets $X_1 = \{x_1, x_2, \ldots, x_{n-1}\}$ and $X_2 = \{x_2, \ldots, x_n\}$ is contained in two sets of the input partitions. W.l.o.g. X_1 is contained in $A_1 \cap B_1$ and X_2 is contained in $A_2 \cap B_2$. Since there are three input partitions and $X_1 \cap X_2$ contains at least one element, then there exists a set $C \in \{A_1, B_1, B_2, A_2\}$ such that $X_1 \subseteq C$ and $X_2 \subseteq C$, and consequently $X \subseteq C$, that is X is contained in a set of at least one of the input partitions. Now if X is contained in a set of two input partitions, then the lemma is verified. Thus assume to the contrary that X is contained in a set of exactly one of the input partitions. This implies that the pair (x_1, x_n) is co-clustered in exactly one input partition, contradicting the assumption that X is a subset of a 2-component. □

An immediate consequence of Lemma 2.1 is that a subset X of U with $|X| \geqslant 2$ is a 2-component iff there exist sets A, B of two partitions such that $X = A \cap B$, which in turn implies that $C_1 \cap C_2$ is the intersection of three sets A, B, C respectively from π_1, π_2, π_3. The 2-components of an instance Π of 3CC have some interesting properties.

Lemma 2.2. *Let X, Y be two 2-components of an instance Π of* 3-CONSENSUS CLUSTERING *such that $|X \cap Y| \geqslant 1$. Then exactly one of the partitions in Π has a set A such that $X \cup Y \subseteq A$, and there are two sets B_1, B_2 of different partitions of Π such that $B_1 \supseteq X$ but $B_1 \not\supseteq Y$ and $B_2 \supseteq X$ but $B_2 \not\supseteq Y$.*

Proof. Since X is a 2-component, by Lemma 2.1 there must exist two sets A, B_1 in different partitions of Π such that X is subset of A and B_1. The same property holds for Y. However since $|X \cap Y| \geqslant 1$ and there are only three partitions in Π, there is a set in a partition containing both X and Y. Note that if there is more than one set containing both X and Y, then $X \cup Y$ is a 2-component, violating the maximality of X and Y. Thus the lemma follows. □

Lemma 2.3. *Given an instance $\Pi = \{\pi_1, \pi_2, \pi_3\}$ of* 3-CONSENSUS CLUSTERING*, let X, Y be two 2-components of Π such that $X \cap Y \neq \emptyset$. Then, all elements in $X \cap Y$ are co-clustered in three partitions. Moreover, there is at most one other 2-component Z sharing elements with both X and Y, and in this case $X \cap Y = X \cap Z = Y \cap Z = X \cap Y \cap Z \neq \emptyset$.*

Proof. By Lemma 2.1, w.l.o.g. we can assume that there exist three subsets A_1, A_2, A_3 of π_1, π_2 and π_3 respectively, such that $X = A_1 \cap A_2$, and $Y = A_2 \cap A_3$. An immediate consequence is that all elements in $X \cap Y$ are co-clustered in three partitions.

Let Z be a 2-component such that $Z \cap X \neq \emptyset$ and $Z \cap Y \neq \emptyset$. Since Z is distinct from X and Y, by the maximality of 2-components and by Lemma 2.1, it follows that $Z = A_1 \cap A_3$.

Applying the first part of the lemma, we know that all elements in the three sets $X \cap Y$, $Y \cap Z$, $X \cap Z$, are co-clustered in all three input partitions, that is they are all included in A_1, A_2, A_3, therefore $A_1 \cap A_2 \cap A_3 \supseteq (X \cap Y) \cup (X \cap Z) \cup (Y \cap Z) \neq \emptyset$. By maximality of 2-components, since all elements in $A_1 \cap A_2 \cap A_3$ are co-clustered in three input partitions, $A_1 \cap A_2 \cap A_3 \subseteq X, Y, Z$ and, *a fortiori*, $A_1 \cap A_2 \cap A_3 \subseteq X \cap Y \cap Z$, hence $X \cap Y \cap Z \supseteq (X \cap Y) \cup (X \cap Z) \cup (Y \cap Z)$. By definition of subset $X \cap Y \cap Z \subseteq X \cap Y$, $X \cap Z$, $Y \cap Z$, hence $X \cap Y \cap Z = X \cap Y = X \cap Z = Y \cap Z$.

It remains to prove that Z is the only 2-component that can possibly share some elements with both X and Y. Assume to the contrary that there exists another such 2-component W. Applying the first part of the lemma we know that $X \cap Y \cap Z = X \cap Y = X \cap Y \cap W$, therefore $Z, W \supseteq X \cap Y \neq \emptyset$. Since W must be included in at least two of A_1, A_2, A_3, assume w.l.o.g. $W \subseteq A_2, A_3$. Since $Z, W \subseteq A_2, A_3$, the maximality of Z and W is violated. □

Corollary 2.4. *Let X, Y, Z be 2-components such that $X \cap Y = X \cap Z = Y \cap Z \neq \emptyset$, then each of $X \cup Y$, $X \cup Z$, $Y \cup Z$ is co-clustered in a distinct partition.*

Proof. Assume that there is a set of a partition, w.l.o.g. π_1, that co-clusters $X \cup Y \cup Z$. Then at least two of X, Y, Z must be co-clustered in one of π_2, π_3 violating the maximality of 2-components. □

A consequence we have the following fundamental property of the 2-components of an instance of MR3CC.

Corollary 2.5. *Let C_1, C_2 be two 2-components of an instance of* MINIMUM RESTRICTED 3-CONSENSUS CLUSTERING*, then $|C_1 \cap C_2| \leqslant 1$.*

A main tool used in the paper is the following graph constructed from the 2-components of an instance of 3-CONSENSUS CLUSTERING.

Definition 2.3. Let Π be an instance of 3-CONSENSUS CLUSTERING and let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be the set of 2-components of Π. Then the *component graph* associated with Π is the graph $G_c = (C, E_c)$ where $(C_i, C_j) \in E_c$ iff $C_i \cap C_j \neq \emptyset$.

3. Inapproximability of Consensus Clustering

In this section, we show the inapproximability of 3-CONSENSUS CLUSTERING, in short 3CC. More precisely we will prove that MR3CC is APX-hard via an L-reduction from the MAXIMUM INDEPENDENT SET problem on cubic

graphs, where we are asked for the largest subset of vertices that are not connected by any edge. MAXIMUM INDEPENDENT SET problem on cubic graphs is known to be APX-hard [2]. This result implies the same inapproximability result also for the case of three (generic) input partitions. Next we give the definition of L-reduction [5].

Definition 3.1 *(L-reduction).* Let π_1, π_2 be two **NPO** problems, with cost functions c_1, c_2 respectively. An L-reduction from π_1 to π_2, $\pi_1 \preccurlyeq_L \pi_2$, is a pair of functions f and g both computable in logarithmic space, such that:

- if x is an instance of π_1 with optimum cost $OPT(x)$, then $f(x)$ is an instance of π_2 with optimum cost $OPT(f(x))$ such that $OPT(f(x)) \leqslant \alpha OPT(x)$, where α is a positive constant;
- if s is a feasible solution of $f(x)$ then $g(s)$ is a feasible solution of x such that $|OPT(x) - c_1(g(s))| \leqslant \beta |OPT(f(x)) - c_2(s)|$, where β is a positive constant.

A relevant property of L-reduction is that it preserves approximability. In particular, let π_1 and π_2 be two **NPO** problems such that π_2 is in **APX** and $\pi_1 \preccurlyeq_L \pi_2$, then also π_1 is in **APX**.

The proof of APX-hardness of MR3CC consists of two separate reductions: the first one is an L-reduction from the MAXIMUM INDEPENDENT SET problem on cubic graphs to the problem of finding a maximum independent set on an artificial class of graphs, called *gadget graphs*, or *G-graphs* in short, the second reduction is an L-reduction from the MAXIMUM INDEPENDENT SET problem on G-graphs to MR3CC.

In particular, the second reduction is based on two main steps: the first step consists of proving that a G-graph associated with a cubic graph is the component graph of an instance of MR3CC, while the second step consists of relating the size of an independent set of a G-graph to the number of co-clustered pairs in a feasible solution of an instance of MR3CC whose component graph is exactly the given G-graph.

3.1. Max independent set is APX-hard on G-graphs

Given a cubic graph $G = (V, E)$, we will associate with G a G-graph $\mathcal{G}$ by constructing for each vertex of G a vertex gadget and for each edge $(v_i, v_j) \in E$, an edge gadget connecting the two vertex gadgets associated with v_i and v_j. For each vertex $v_i \in V$, the vertex gadget VG_i is the graph represented in Fig. 1.

Since, in a cubic graph, a vertex v_i is adjacent to three edges, the vertex gadget VG_i has three vertices, $c_{i,1}, c_{i,4}, c_{i,12}$, called *docking vertices*, each one connecting VG_i to another vertex gadget through an edge gadget associated with an edge incident to v_i. We denote by $VG(\mathcal{G})$ the set of vertex gadgets associated with the vertex set of graph G.

Given two adjacent vertices $v_i, v_j \in V$ and the corresponding vertex gadgets VG_i and VG_j, respectively, there is an edge gadget $EG_{i,j}$ associated with the edge (v_i, v_j). The edge gadget $EG_{i,j}$ is the graph of 6 vertices joining VG_i, VG_j in two of their docking vertices c_{i_k}, c_{j_l} with $k, l \in \{1, 4, 12\}$ (see Fig. 2).

Two vertex gadgets are said *independent* if there is no edge gadget between them; otherwise they are *adjacent*. It is easy to note that each vertex gadget VG_i has a unique maximum independent set of cardinality 6 ($\{c_{i,1}, c_{i,4}, c_{i,5}, c_{i,8}, c_{i,9}, c_{i,12}\}$) and that all the docking vertices of VG_i are part of this set (see Fig. 3). We denote this independent set of a vertex gadget as *type* 1. There are two independent sets of VG_i having cardinality 5 and no docking vertices, for example ($\{c_{i,2}, c_{i,3}, c_{i,6}, c_{i,10}, c_{i,11}\}$). We denote this independent set of a vertex gadget as *type* 2. Observe that given a maximum independent set of $\mathcal{G}$ no two adjacent vertex gadgets have both an independent set of type 1. This fact allows us to relate the sizes of maximum independent sets in cubic graphs and G-graphs. We call an independent set

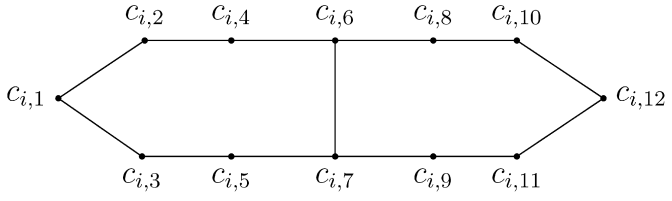


Fig. 1. A vertex gadget VG_i.

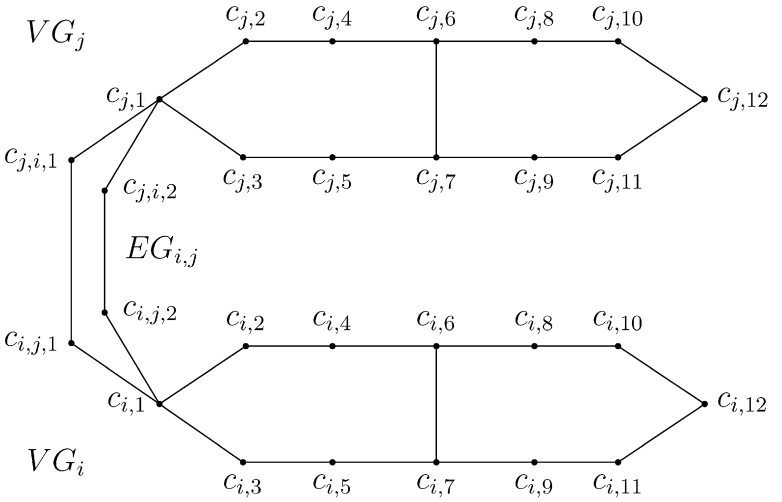


Fig. 2. Vertex gadgets VG_i, VG_j and the edge gadget EG_{ij}.

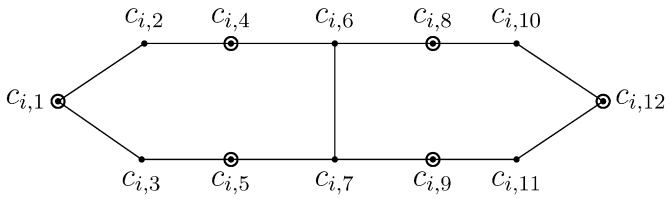


Fig. 3. Maximum Independent Set of a vertex gadgets VG_i.

of $\mathcal{G}$ a *canonical solution* if all vertex gadgets have an independent set of type 1 or type 2 and each edge gadget $EG_{i,j}$ has exactly two of its non-docking vertices in the independent set.

A simple observation on the possible independent sets of an edge gadget $EG_{i,j}$ allows to derive the following important property.

Observation 3.1. *Let $\mathcal{G}$ be a G-graph associated with a cubic graph, let $EG_{i,j}$ be an edge gadget incident on vertex gadgets VG_i, VG_j and let I be an independent set of $\mathcal{G}$. Then I contains at most 2 vertices of $EG_{i,j}$ that are not docking vertices, and if both docking vertices of $EG_{i,j}$ (i.e. vertices of the vertex gadgets) are in I, then no other vertex of $EG_{i,j}$ is in the independent set.*

Lemma 3.2. *Let $\mathcal{G}$ be the G-graph associated with a cubic graph $G = (V, E)$ and let I be an independent set of $\mathcal{G}$ that is not a canonical solution. Then it is possible to compute in polynomial time a canonical solution of size at least $|I|$.*

Proof. Notice that each vertex gadget has an independent set of size 5 that is of type 2 and does not include the docking vertices, that is it does not include the vertices of the edge gadget. Therefore for each vertex gadget for which I does not induce an independent set of type 1 or type 2, replace its independent set with one of type 2. The resulting independent set is at least as large as the original one.

Let $EG_{i,j}$ be a generic edge gadget. If in I both vertex gadgets VG_i and VG_j have an independent set of type 1, that is all their docking vertices are in the independent set I, by Observation 3.1 no other vertex of the edge gadget $EG_{i,j}$ can be in the independent set I. Let us construct a new independent set I^* from I as follows. Update the independent set of VG_i in I so that VG_i becomes of type 2 and vertices $c_{i,j,1}$, $c_{i,j,2}$ of the edge gadget $EG_{i,j}$ are added to I^*. It is immediate to note that $|I^*| = |I| + 1$, as I^* contains a type 1 and type 2 independent set of VG_i and VG_j respectively, and two vertices for $EG_{i,j}$, thus 13 vertices, while the former contains two independent sets of type 1, thus 12 vertices.

Applying the above procedure to each edge gadget of $\mathcal{G}$ leads to a canonical solution whose size is not smaller than I. □

The following theorem shows the effects of our reduction from the independent set problem on cubic graphs to the same problem on G-graphs.

Theorem 3.3. *Let $G = (V, E)$ be a cubic graph with $|E| = m$, $|V| = n$, and let $\mathcal{G}$ be the G-graph associated with G. Then G has an independent set of size at least h if and only if $\mathcal{G}$ has an independent set of size at least $6h + 5(n - h) + 2m$.*

Proof. Let I be an independent set of G, with $|I| \geqslant h$. Let I_c be the independent set of $\mathcal{G}$ obtained by imposing that the independent set of the vertex gadget VG_i is of type 1 if $v_i \in I$, otherwise the independent set of VG_i is of type 2. Moreover assume that each edge gadget has in I_c two vertices that are not docking vertices. By an immediate counting argument I_c is an independent set of $\mathcal{G}$ of size $6h + 5(n - h) + 2m$.

Now let I_c be an independent set of graph $\mathcal{G}$ of size at least $6h + 5(n - h) + 2m$. First observe that by Lemma 3.2, we can compute a canonical solution I of size at least $|I_c|$. Since I is canonical, each edge gadget must have exactly two non-docking vertices in the independent set I. Hence in I there exist h vertex gadgets with an independent set of type 1, $n - h$ vertex gadgets with an independent set of type 2. We already know that two adjacent vertex gadgets cannot both have an independent set of type 1 in I, therefore the set of h vertex gadgets associated with an independent set of type 1 in I identifies an independent set of G with size h. has an independent set of type 1 in I. □

Since for every cubic graph $G = (V, E)$, $|E| = \frac{3}{2}|V|$ and there exists an independent set of size at least $|V|/4$, the reduction stated in Theorem 3.3 is actually an L-reduction from cubic graphs to the class of G-graphs. Consequently computing the maximum independent set is APX-hard also for G-graphs.

3.2. Reducing MIS on G-graphs to MR3CC

In this section we build an L-reduction from MIS on G-graphs to MR3CC. The reduction consists of two basic steps. The first basic step of our second reduction is to build from a G-graph associated with a cubic graph an instance of MR3CC whose component graph is the given G-graph. Hence, given a G-graph $\mathcal{G}$, we first define the 2-components for each vertex gadget and for each edge gadget associated with $\mathcal{G}$. Thus we prove in Lemma 3.4 how to construct an instance Π of MR3CC such that the component graph associated with Π is $\mathcal{G}$.

The second basic step of the reduction relates the size of an independent set in $\mathcal{G}$ to the cost of a feasible solution to the MR3CC instance constructed in the first step from $\mathcal{G}$. More precisely, we will prove that we can focus on a special class of solutions of MR3CC, called *normal solutions*, and we will prove various properties of this kind of solutions that will be useful to establish the relationship between sizes of solutions in the reduction.

In the following, we first associate with each vertex and edge gadget a set of 2-components of a set Π consisting of three partitions. Consider a vertex gadget VG_i and the corresponding 2-components represented in Fig. 4.

Associated with VG_i there is a set Π of three partitions over the set $U_i = \{i_1, i_2, \ldots, i_{35}\}$. The partitions of Π and the 2-components are the ones of Fig. 4:

- $\pi_1(VG_i) = (c_{i,1} \cup c_{i,2} \cup c_{i,4}, c_{i,5} \cup c_{i,7}, c_{i,8}, c_{i,11} \cup c_{i,12}, \{i_8\}, \{i_9\}, \{i_{17}\}, \{i_{26}\}, \{i_{27}\}, \{i_{28}\}, \{i_{29}\})$,

$c_{i,1} = \{i_1, i_2, i_3, i_4\}$
$c_{i,2} = \{i_1, i_5, i_6, i_7\}$
$c_{i,3} = \{i_2, i_8, i_9, i_{10}\}$
$c_{i,4} = \{i_7, i_{11}, i_{12}, i_{13}\}$
$c_{i,5} = \{i_{10}, i_{14}, i_{15}, i_{16}\}$
$c_{i,6} = \{i_{13}, i_{17}, i_{18}, i_{19}\}$
$c_{i,7} = \{i_{14}, i_{19}, i_{20}, i_{21}\}$
$c_{i,8} = \{i_{18}, i_{22}, i_{23}, i_{24}\}$
$c_{i,9} = \{i_{20}, i_{26}, i_{27}, i_{25}\}$
$c_{i,10} = \{i_{22}, i_{28}, i_{29}, i_{30}\}$
$c_{i,11} = \{i_{25}, i_{31}, i_{32}, i_{33}\}$
$c_{i,12} = \{i_{30}, i_{33}, i_{34}, i_{35}\}$

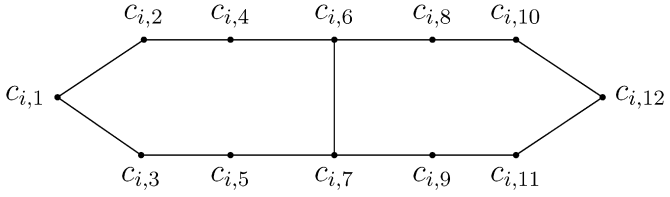


Fig. 4. A vertex gadget and its 2-components.

- $\pi_2(VG_i) = (c_{i,1} \cup c_{i,3}, c_{i,4} \cup c_{i,6} \cup c_{i,7} \cup c_{i,9}, c_{i,10} \cup c_{i,12}, \{i_5\}, \{i_6\}, \{i_{15}\}, \{i_{16}\}, \{i_{23}\}, \{i_{24}\}, \{i_{31}\}, \{i_{32}\})$,
- $\pi_3(VG_i) = (c_{i,2}, c_{i,3} \cup c_{i,5}, c_{i,6} \cup c_{i,8} \cup c_{i,10}, c_{i9} \cup c_{i,11}, \{i_3\}, \{i_4\}, \{i_{11}\}, \{i_{12}\}, \{i_{21}\}, \{i_{34}\}, \{i_{35}\})$.

Observe that each set $c_{i,z}$ is a 2-component of Π. Indeed, the elements of each set $c_{i,z}$ are co-clustered by construction in two partitions. Now consider two sets X, Y of partitions π_a, π_b respectively, with $1 \leqslant a < b \leqslant 3$. Observe that either $X \cap Y = c_{i,u}$, with $1 \leqslant u \leqslant 12$, or $X \cap Y = i_v$, with $1 \leqslant v \leqslant 35$, or $X \cap Y = \emptyset$. Now, if $X \cap Y = c_{i,u}$, since $c_{i,u}$ is co-clustered in X and Y with disjoint sets of elements it follows that $c_{i,u}$ is a 2-component. Furthermore, observe that by construction for each $c_{i,u}$ there are at most two sets X, Y in π_1, π_2, π_3 such that $X \cap Y = c_{i,u}$.

Consider $c_{i,1}$, $c_{i,4}$, $c_{i,12}$ the docking vertices of vertex VG_i. Note that each of them shares two elements with some other 2-components of the same vertex gadget, while two elements are not shared with any other 2-component of the vertex gadgets, let us call these two unshared elements *private*. In particular the private elements of $c_{i,1}$ are i_3, i_4, those of $c_{i,4}$ are i_{11}, i_{12} and those of $c_{i,12}$ are i_{34}, i_{35}. Denote by $d_i(EG_{i,j})$, $d_j(EG_{i,j})$ the docking vertices $c_{i,k}$, $c_{j,l}$ of the edge gadget $EG_{i,j}$, where $k, l \in \{1, 4, 12\}$. Moreover, denote by $p_1(d_i(EG_{i,j}))$, $p_2(d_i(EG_{i,j}))$ the two private elements of the docking vertex $d_i(EG_{i,j})$. We can then describe the 2-components associated with vertices of $EG_{i,j}$. These 2-components are vertices $c_{i,j,1} = \{p_1(d_i(EG_{i,j})), h_{i,j,1}, e_{i,j,1}\}$, $c_{i,j,2} = \{p_2(d_i(EG_{i,j})), e_{i,j,2}\}$, $c_{j,i,1} = \{p_1(d_j(EG_{i,j})), h_{i,j,2}, e_{i,j,1}\}$ and $c_{j,i,2} = \{p_2(d_j(EG_{i,j})), e_{i,j,2}\}$.

We still have to show how to construct the three partitions π_1, π_2, π_3 associated with a G-graph $\mathcal{G}$. Observe that for each pair VG_i, VG_j of vertex gadgets, the three partitions we associate with VG_i are over a universe set U_i that is disjoint from the universe set U_j of the three partitions of VG_j. Consequently, we can construct three partitions associated with all vertex gadgets as the union of the partitions for each single gadget. Formally, for each $j \in \{1, 2, 3\}$, then $\pi_j(VG(\mathcal{G})) = \bigcup_{vg \in VG(\mathcal{G})} \pi_j(vg)$. Furthermore, other sets "producing" the 2-components associated with edge gadgets are appropriately added to the three partitions $\pi_1(VG(\mathcal{G}))$, $\pi_2(VG(\mathcal{G}))$ and $\pi_3(VG(\mathcal{G}))$ to get π_1, π_2, π_3.

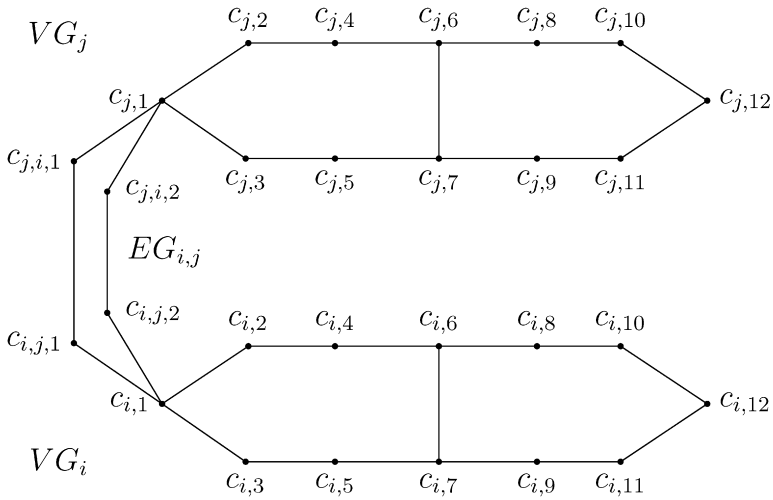
The procedure for computing the partitions associated with a G-graph is Algorithm 1, which clearly requires polynomial time. In Algorithm 1, the procedure $Add(\pi, a, X)$ computes the partition obtained from π by adding all elements in $X - \{a\}$ to the set of π to which a belongs. Observe that at line 10 and 11 we merge two 2-components and add the resulting set to π_3. Indeed consider w.l.o.g. $c_{i,j,1}$ and $c_{j,i,1}$. Before line 10, the elements of $c_{i,j,1}$ are co-clustered only in a set of partition π_1, while they are not co-clustered in π_2 (see lines 4–9 of the algorithm). A similar property holds for $c_{j,i,1}$, since the elements of $c_{j,i,1}$ are co-clustered only in a set of partition π_2, while they are not co-clustered in π_1 . Now, $c_{i,j,1}$, $c_{j,i,1}$ must be contained in a set of π_3 and, since $c_{i,j,1} \cap c_{j,i,1} = \{e_{i,j,1}\}$, they must belong to the same set of π_3.

Algorithm 1. Reduce($\mathcal{G}$)

1. **Input:** a G-graph $\mathcal{G}$
2. **For each** $l = 1$ to 3 $\pi_l(VG(\mathcal{G})) = \bigcup_{vg \in VG(\mathcal{G})} \pi_l(vg)$
3. **For each** edge gadget $EG_{i,j}$ incident on VG_i and VG_j **do**
4. $\quad \pi_1 \leftarrow Add(\pi_1, p_1(d_i(EG_{i,j})), c_{i,j,1})$
5. $\quad \pi_1 \leftarrow Add(\pi_1, p_2(d_j(EG_{i,j})), c_{j,i,2})$
6. $\quad$ add to π_1 all elements of $c_{i,j,2} \cup c_{j,i,1}$ as singletons
7. $\quad \pi_2 \leftarrow Add(\pi_2, p_2(d_i(EG_{i,j})), c_{i,j,2})$
8. $\quad \pi_2 \leftarrow Add(\pi_2, p_1(d_j(EG_{i,j})), c_{j,i,1})$
9. $\quad$ add to π_2 all elements of $c_{i,j,1} \cup c_{i,j,2}$ as singletons
10. $\quad \pi_3 \leftarrow Add(\pi_3, p_1(d_i(EG_{i,j})), c_{i,j,1} \cup c_{j,i,1})$
11. $\quad \pi_3 \leftarrow Add(\pi_3, p_2(d_i(EG_{i,j})), c_{i,j,2} \cup c_{j,i,2})$
12. **Return**(π_1, π_2, π_3)

In Fig. 5 we show vertex gadgets VG_i, VG_j and edge gadget $EG_{i,j}$ with the corresponding 2-components.
Then, we can show that Algorithm 1 correctly computes an instance of MR3CC.

Lemma 3.4. *Let $\mathcal{G}$ be a G-graph. Then Algorithm* 1 *computes a set Π of three partitions π_1, π_2, π_3 such that $\mathcal{G}$ is the component graph of Π, and Π is an instance of* MR3CC *associated with $\mathcal{G}$.*



$c_{j_1} = \{j_1, j_2, j_3, j_4\}$, $c_{j_2} = \{j_1, j_5, j_6, j_7\}$, $c_{j_3} = \{j_2, j_8, j_9, j_{10}\}$, $c_{j_4} = \{j_7, j_{11}, j_{12}, j_{13}\}$, $c_{j_5} = \{j_{10}, j_{14}, j_{15}, j_{16}\}$, $c_{j_6} = \{j_{13}, j_{17}, j_{18}, j_{19}\}$, $c_{j_7} = \{j_{14}, j_{19}, j_{20}, j_{21}\}$, $c_{j_8} = \{j_{18}, j_{22}, j_{23}, j_{24}\}$, $c_{j_9} = \{j_{20}, j_{26}, j_{27}, j_{25}\}$, $c_{j_{10}} = \{j_{22}, j_{28}, j_{29}, j_{30}\}$, $c_{j_{11}} = \{j_{25}, j_{31}, j_{32}, j_{33}\}$, $c_{j_{12}} = \{j_{30}, j_{33}, j_{34}, j_{35}\}$ $c_{i,1} = \{i_1, i_2, i_3, i_4\}$, $c_{i,2} = \{i_1, i_5, i_6, i_7\}$, $c_{i,3} = \{i_2, i_8, i_9, i_{10}\}$, $c_{i,4} = \{i_7, i_{11}, i_{12}, i_{13}\}$, $c_{i,5} = \{i_{10}, i_{14}, i_{15}, i_{16}\}$, $c_{i,6} = \{i_{13}, i_{17}, i_{18}, i_{19}\}$, $c_{i,7} = \{i_{14}, i_{19}, i_{20}, i_{21}\}$, $c_{i,8} = \{i_{18}, i_{22}, i_{23}, i_{24}\}$, $c_{i,9} = \{i_{20}, i_{26}, i_{27}, i_{25}\}$, $c_{i,10} = \{i_{22}, i_{28}, i_{29}, i_{30}\}$, $c_{i,11} = \{i_{25}, i_{31}, i_{32}, i_{33}\}$, $c_{i,12} = \{i_{30}, i_{33}, i_{34}, i_{35}\}$ $c_{j,i,1} = \{j_3, h_{i,j,2}, e_{i,j,1}\}$, $c_{j,i,2} = \{j_4, e_{i,j,2}\}$, $c_{i,j,1} = \{i_3, h_{i,j,1}, e_{i,j,1}\}$, $c_{i,j,2} = \{i_4, e_{i,j,2}\}$.

Fig. 5. Vertex gadgets VG_i, VG_j, edge gadget EG_{ij} and the corresponding 2-components.

Proof. We prove the lemma by induction on the number of edge gadgets in $\mathcal{G}$. If $\mathcal{G}$ has no edge gadget, the lemma holds since line 2 of the algorithm builds the correct partitions.

Consider now a component graph $\mathcal{G}$ with m edge gadgets; removing the edge gadget $EG_{i,j}$ leads to a component graph $\mathcal{G}'$ with $m-1$ edge gadgets, for which the lemma holds by inductive hypothesis (actually for such a new graph some docking vertices are not used by any edge gadgets, but this fact is only a minor annoyance). Consequently the partitions computed by the algorithm are correct, and let $\pi_1^1, \pi_2^1, \pi_3^1$ be such partitions.

Now we analyze the situation after the instructions at lines 4–11. Remember that the docking vertices of $EG_{i,j}$ are $d_i(EG_{i,j})$ and $d_j(EG_{i,j})$. By construction the elements in $c_{i,j,1}$ are co-clustered in π_1, π_3, and similarly, also the elements in $c_{j,i,2}$ are co-clustered in π_1, π_3. By construction it follows that all the elements in $c_{j,i,1}$ are co-clustered in π_2, π_3 and similarly, also the elements in $c_{i,j,2}$ are co-clustered in π_2, π_3.

Note that the only elements of the vertex gadgets that are co-clustered in two partitions with elements of $EG_{i,j}$ are the private elements of the docking vertices of $EG_{i,j}$. This fact follows from the observation that at line 9, that is before the execution of the instructions at line 10 and 11, each private element of a docking vertex is a singleton in π_3. Moreover note that $c_{i,j,1} \cap c_{j,i,1} = \{e_{i,j,1}\}$ and $c_{i,j,2} \cap c_{j,i,2} = \{e_{i,j,2}\}$. Now consider w.l.o.g. the instruction at line 10. Since $c_{i,j,1} \cap c_{j,i,1} = \{e_{i,j,1}\}$, it follows that after adding $c_{i,j,1} \cup c_{j,i,1}$ to π_3 there is no element of $c_{i,j,1} \setminus \{e_{i,j,1}\}$ co-clustered in two partitions with an element $c_{j,i,1} \setminus \{e_{i,j,1}\}$. Hence $c_{i,j,1}$, $c_{j,i,1}$ are maximal and are 2-components. Similarly, we can prove that $c_{i,j,2}$ and $c_{j,i,2}$ are 2-components.

Observe that each 2-component of VG_i and VG_j is co-clustered with a 2-component of $EG_{i,j}$ in at most one partition. Hence after the execution of the instructions at lines 4–11 of the algorithm, each 2-component of VG_i, VG_j preserves the maximality property and hence it is a 2-component.

Finally note that distinct edge gadgets do not share elements, and thus each iteration of lines 4–11 can be applied independently to each edge gadget. This fact completes the proof. □

The second basic step of the reduction from the Maximum Independent Set problem on a G-graph $\mathcal{G}$ to MR3CC consists in relating the cost of a feasible solution to the instance of MR3CC associated with $\mathcal{G}$ to the 2-components or vertices of $\mathcal{G}$.

To this end, we will prove several technical results showing that an arbitrary feasible solution π of an instance $\Pi = \{\pi_1, \pi_2, \pi_3\}$ of MR3CC associated with a G-graph $\mathcal{G}$ can be iteratively modified in polynomial time so that the resulting solution π^* is in a special form allowing for an easy identification of an independent set of the G-graph.

In our proofs, given an edge gadget $EG_{i,j}$, we denote by $P(EG_{i,j})$ the set of pairs that are co-clustered in the 2-components $c_{i,j,1}$, $c_{i,j,2}$, $c_{j,i,1}$, $c_{j,i,2}$. Given an edge gadget VG_i, we denote by $P(VG_i)$ the set of pairs that are co-clustered in the 2-components of VG_i, $c_{i,1}$, $c_{i,2}$, …, $c_{i,12}$. We are now able to introduce the notion of normal solution; we will then see that only normal solutions have to be taken into account in all our proofs.

Definition 3.2 *(Normal solution).* Let π be a solution for an instance Π of MR3CC. Then π is *normal* if:

(1) each set in π is a (not necessarily proper) subset of a 2-component of the component graph associated with Π,
(2) for each 2-component C, at most one subset of C is a set in π.

A normal solution π can be described as a collection of disjoint subsets of the 2-components; in fact we will denote by $S_\pi(C)$ the subset of the 2-component C that is a set of the solution π. Whenever possible we will drop the subscript denoting the normal solution. Notice that some of the sets $S_\pi(C)$ can be empty.

A normal solution π is a *type* (a) *solution* for the vertex gadget VG_i (or VG_i is *a-induced* by π—we will drop the solution π whenever ambiguities do not arise) if π is a normal solution and the 2-components $c_{i,1}$, $c_{i,4}$, $c_{i,5}$, $c_{i,8}$, $c_{i,9}$, $c_{i,12}$ are all sets of π. Moreover π is a *type* (b) solution for the vertex gadget VG_i (or VG_i is *b-induced* by π) if π is a normal solution and the 2-components $c_{i,2}$, $c_{i,3}$, $c_{i,10}$, $c_{i,11}$, and one of either $c_{i,6}$ or $c_{i,7}$ are all sets of π. A normal solution π is *canonical* if each vertex gadget is a-induced or b-induced and π co-clusters exactly 4 pairs of $P(EG_{i,j})$, for each edge gadget. In such case we will also show that the set of a-induced vertex gadgets correspond to a set of mutually independent vertex gadgets, completing our reduction. The main idea of our proof is that a normal solution π can induce only a few specific solutions in each vertex gadget, and that a canonical solution can be obtained from any solution in polynomial time.

Lemma 3.5. *Let π be a solution for an instance Π of* MR3CC. *Then, there exists a normal solution π^* for Π whose cost is no greater than the cost of π.*

Proof. In the following we will show that Algorithm 2 updates a partition π without ever increasing its cost. This fact, together with proving that the partition returned by Algorithm 2 is normal, suffices to prove the lemma.

Notice that when line 9 of Algorithm 2 is reached, all sets in $\mathcal{D}$ are marked. Since a set can be marked only at line 6, all sets in $\mathcal{D}$ are contained in a 2-component of the input partition. In lines 9–10, all sets that are contained in a same 2-component are merged together, therefore in the final partition no two sets are included in the same 2-component. Consequently, the partition returned by Algorithm 2 is normal.

Given Π the instance of MR3CC, then the cost of the solution π can be expressed as $c(\pi) = \sum_{i,j \in U} c_\pi(i,j)$, where U is the universe set of π and by Eq. (1), $c_\pi(i,j) = d_\Pi(i,j)$ if i, j are co-clustered in π or $c_\pi(i,j) = s_\Pi(i,j)$ if i, j are not co-clustered in π. Observe that for each pair (i,j) co-clustered in π and contained in a 2-component of Π, $c_\pi(i,j) = 1$. If i, j are co-clustered in π and are not contained in a 2-component of Π, then $c_\pi(i,j) \geqslant 2$, as the pair (i,j) is not co-clustered in either 2 or 3 input partitions.

We will begin by analyzing lines 1–8 of Algorithm 2, which proceeds by updating a partition π solution of MR3CC over instance Π as follows: a set $X \in \pi$ is split into two subsets D and $X - D$ (where $D \in \mathcal{D}$ is a subset of a 2-component) in lines 5–8; no other part modifies the partition π. Alternatively we can view a single iteration of lines 5–8 of the algorithm as taking some co-clustered pairs and transforming them into not co-clustered pairs, hence the difference between the costs of the partition before and after the execution of lines 5–8 is due to two sets of pairs of elements in X, P and $\bar{P}$, that are co-clustered before the execution but not after. More precisely P is the set of pairs of elements in X that are in 2-components and that are co-clustered before the execution but not after, and $\bar{P}$ is the set of pairs of elements in X that are not in 2-components and are co-clustered before the execution but not after.

Clearly for each pair $(x,y) \in P$, the cost before the execution is 1 and the cost after the execution is 2, while for each pair $(x,y) \in \bar{P}$ the reverse is true, that is the cost before the execution is at least 2 and the cost after the execution is 1, consequently $|\bar{P}| - |P|$ is a lower bound on the improvement made by a single execution of lines 5–8.

An immediate consequence is that proving $|\bar{P}| \geqslant |P|$ would complete the proof that lines 1–8 do not increase the cost of the partition.

We are interested only in the executions of lines 5–8 where π is actually modified, that is X is split into two non-empty sets. Note that for each pair $(x, y) \in P$, there exists a 2-component C of Π containing both x and y. Let us denote $D' = C \cap X$, moreover w.l.o.g. $x \in D' \cap D$ and $y \in D' - D$, since $(x, y) \in P$, and consequently $D \neq D'$. Notice also that $D' \in \mathcal{D}$, by construction of $\mathcal{D}$. Clearly, $|D' \cap D| = 1$ by Corollary 2.5 which, together with Lemma 2.3, immediately implies that (x, y) is the only pair in P containing y.

We claim that for each $(x, y) \in P$ there exists a pair $(z, y) \in \bar{P}$, where $z \in D - D'$. Since (x, y) is the only pair in P containing y, this fact implies that $|\bar{P}| \geqslant |P|$, as required. We know that $x, y \in D'$, hence $|D'| \geqslant 2$, while $|D| \geqslant |D'|$, by the choice of D at line 5 of Algorithm 2. Since $D \cap D' = \{x\}$, there exists $z \in D - D'$. It is immediate to note that $(z, y) \in \bar{P}$, hence we have not increased the cost of the partition. Also notice that, by construction, each set of π is included in some 2-component.

We have to prove that lines 9–10 do not increase the cost of π. The effect of lines 9–10 is to co-cluster some pairs that were previously not co-clustered. All such pairs are contained in the same 2-component, therefore the total cost is not increased. □

Algorithm 2. Proof of Lemma 3.5

1. **Input:** a solution π, $\mathcal{C} \leftarrow$ set of 2-components of Π
2. **While** there exists an unmarked $X \in \pi$ **do**
3. $\quad \mathcal{D} \leftarrow \{C \cap X\colon\ C \in \mathcal{C} \wedge C \cap X \neq \emptyset\}$
4. $\quad$ **While** $\mathcal{D} \neq \emptyset$ **do**
5. $\quad\quad D \leftarrow$ the set in $\mathcal{D}$ of maximum size
6. $\quad\quad$ Mark D and add it to π
7. $\quad\quad X \leftarrow X \setminus D$
8. $\quad\quad$ Update $\mathcal{D} \leftarrow \{C \cap X\colon\ C \in \mathcal{C} \wedge C \cap X \neq \emptyset\}$
   $\quad\quad$ //Equivalently $\mathcal{D} \leftarrow \{X_{\mathcal{D}} \setminus D\colon\ X_{\mathcal{D}} \in \mathcal{D} \wedge X_{\mathcal{D}} \setminus D \neq \emptyset\}$
9. **ForEach** 2-component C
10. $\quad$ Merge all sets of π that are contained in C
11. **Return** π

Let us recall that the cost $c(\pi)$ of an optimal solution π of MR3CC is given by Eq. (1), where $0 \leqslant s_\Pi(i, j) \leqslant 2$. Recall that $r_\pi(i, j) = 1$ if elements i and j are co-clustered in the solution π, $r_\pi(i, j) = 0$ otherwise. From (1) the cost of a solution π, $c(\pi) = \sum_{i<j} s_\Pi(i, j) + \sum_{i<j} r_\pi(i, j)(d_\Pi(i, j) - s_\Pi(i, j))$. First, observe that the term $s_\Pi(i, j)$ in $c(\pi)$ does not depend from π. For a normal solution to a MR3CC instance, $r_\pi(i, j) = 1$ implies $s_\Pi(i, j) = 2$ and $d_\Pi(i, j) = 1$. Hence the cost of a normal solution $c(\pi)$ is equal to $\sum_{i<j} s_\Pi(i, j) - |\{(i, j)\colon\ i < j \text{ and } r_\pi(i, j) = 1\}|$. By Lemma 3.5 there is always an optimal normal solution π, consequently, the cost $c(\pi)$ is minimized when is maximized the number of co-clustered pairs in π. We say that a normal solution π *covers* a 2-component C, or *has* a 2-component C, if $C \in \pi$.

Lemma 3.6. *Given a component graph G_c associated with an instance $\Pi = \{\pi_1, \pi_2, \pi_3\}$ of* MR3CC*, let π be a normal solution and let C be the set of 2-components of Π covered by π. Then C is an independent set of G_c.*

Proof. Assume to the contrary that π covers two 2-components C_1 and C_2 of Π corresponding to adjacent vertices of graph G_c. Then, by construction of the graph G_c, C_1 and C_2 share a common element, contradicting the assumption that π is a partition. □

The following propositions (up to Lemma 3.15) regard constant-size structures, such as vertex and edge gadgets, and can be verified by enumerating all possibilities. Nonetheless we will give the formal proofs in order to provide some insights on the reduction.

For simplicity's sake, we assume in the following that Π is an instance of MR3CC associated with a G-graph. Moreover, we denote by $\Pi_i = \{\pi_1(VG_i), \pi_2(VG_i), \pi_3(VG_i)\}$ the partitions induced in Π by the vertex gadget VG_i.

Analogously, given an edge gadget $EG_{i,j}$, we denote by $\Pi_{i,j} = \{\pi_1(VG_i \cup VG_j \cup EG_{i,j}), \pi_2(VG_i \cup VG_j \cup EG_{i,j}), \pi_3(VG_i \cup VG_j \cup EG_{i,j})\}$ the partitions induced in Π by the vertex gadgets VG_i, VG_j and the edge gadget $EG_{i,j}$.

Lemma 3.7. *Given Π and a vertex gadget VG_i, let c_x, c_y, c_z be 2-components of Π_i such that c_y, c_z are the only 2-components adjacent to c_x. Let π be a normal solution for instance Π_i, then the following two properties hold*:

(1) *if π does not cover any of c_x, c_y, c_z, then there exists a normal solution π' for Π_i that covers c_x and π' co-clusters more pairs than π;*
(2) *if π covers c_y and does not cover c_x and c_z, then there exists a normal solution π' that covers c_x and π' co-clusters at least as many pairs than π.*

Proof. Recall that, given a 2-component c_x, we denote by $S_\pi(c_x)$ the subset of the 2-component c_x that is a set of the solution π. By definition of normal solution, $c_x - (c_y \cup c_z) \subseteq S_\pi(c_x)$. Moreover by Corollary 2.5 the sets $c_x \cap c_y$ and $c_x \cap c_z$ consist of one element each, let us denote by xy and xz respectively such elements. Since c_x is not covered in π, at least one of xy and xz does not belong to $S_\pi(c_x)$. W.l.o.g. we can assume that $xy \notin S_\pi(c_x)$.

In what follows we will build from π a new normal solution π' that covers c_x, by moving elements xy and xz in $S_\pi(c_x)$, if they do not already belong to $S_\pi(c_x)$.

First let us prove statement (1) of the lemma, that is assume that π does not cover c_x, c_y, c_z. If $xz \notin S_\pi(c_x)$, since we have assumed that $xy \notin S_\pi(c_x)$, clearly $|S_\pi(c_x)| = 2$. Since $xz \notin S_\pi(c_x)$, $xz \in S_\pi(c_z)$ and $|S_\pi(c_z)| = 3$ as c_z is not covered by π and π is normal. Therefore eventually moving xz into $S_{\pi'}(c_x)$ does not decrease the number of co-clustered pairs, since $|S_{\pi'}(c_x)| = 3$ and $|S_{\pi'}(c_z)| = 2$. Consequently we can assume that $xz \in S_\pi(c_x)$.

Since $xy \notin S_\pi(c_x)$, there are 3 co-clustered pairs in $S_\pi(c_x)$, as $|S_\pi(c_x)| = 3$, while xy belongs to the set $S_\pi(c_y)$. Since c_y is not covered by π, it follows that $|S_\pi(c_y)| = 3$ and there are 3 co-clustered pairs in $S_\pi(c_y)$. Now, move xy into $S_{\pi'}(c_x)$. It follows that $|S_{\pi'}(c_x)| = 4$ and hence there are 6 co-clustered pairs in $S_{\pi'}(c_x)$, while $|S_{\pi'}(c_y)| = 2$ and hence there is 1 co-clustered pairs in $S_{\pi'}(c_y)$ Thus the resulting solution covers c_x and increases the number of co-clustered pairs.

Now let us prove statement (2), that is assume that π covers c_y. If $xz \notin S_\pi(c_x)$, then move xz into $S_{\pi'}(c_x)$. Since c_z is not covered in π, the element xz is co-clustered with at most two other elements of c_z, while $|S_\pi(c_x)| = 2$. It follows immediately that moving xz into $S_{\pi'}(c_x)$ does not decrease the number of co-clustered pairs. Now move xy into $S_{\pi'}(c_x)$. Clearly xy in π is co-clustered with three other elements, since c_y is covered by π. Similarly, xy in $S_{\pi'}(c_x)$ is co-clustered with three other elements of c_x, implying that moving xy into $S_{\pi'}(c_x)$ does not decrease the number of co-clustered pairs. The resulting partition π' covers c_x, hence completing the proof. □

Lemma 3.8. *Given Π and a vertex gadget VG_i which is a-induced by the normal solution π, π co-clusters* 41 *pairs of $P(VG_i)$.*

Proof. All normal type (a) solutions for VG_i have the following sets, by definition of a-induced and of normal solution: $c_{i,1}$, $c_{i,4}$, $c_{i,5}$, $c_{i,8}$, $c_{i,9}$, $c_{i,12}$, $c_{i,2} \setminus (c_{i,1} \cup c_{i,4})$, $c_{i,3} \setminus (c_{i,1} \cup c_{i,5})$, $c_{i,6} \setminus (c_{i,4} \cup c_{i,8})$, $c_{i,7} \setminus (c_{i,5} \cup c_{i,6} \cup c_{i,9})$, $c_{i,10} \setminus (c_{i,8} \cup c_{i,12})$, $c_{i,11} \setminus (c_{i,9} \cup c_{i,12})$. It is immediate to notice that the number of co-clustered pairs is $6 \cdot 6 + 5 \cdot 1 = 41$. □

Observe that there is a clear relationship between an a-induced vertex gadget and a type 1 independent set for VG_i. Indeed, a type 1 independent set of VG_i consists of nodes $c_{i,1}$, $c_{i,4}$, $c_{i,5}$, $c_{i,8}$, $c_{i,9}$, $c_{i,12}$: these are precisely the 2-components that are covered by π.

Now we have to analyze the structure of b-induced vertex gadgets. Let π be a normal solution over instance Π. For each docking vertex d of VG_i, if both private elements of d are in $S_\pi(d(VG_i))$, that is they are co-clustered in π, we will say that π is *strict* for d. If all the docking vertices of a vertex gadget VG_i are strict in a solution π, we say that π is strict for VG_i. A solution π is strict type (b) for VG_i (respectively strict type (a) for VG_i) if VG_i is b-induced (respectively a-induced) in π and all the docking vertices of VG_i are strict.

Lemma 3.9. *Given Π, a vertex gadget VG_i, and a normal solution π that is strict type* (b) *for VG_i, the number of pairs of $P(VG_i)$ co-clustered in π is equal to* 40.

Proof. Given π a strict type (b) solution for VG_i then $c_{i,2}$, $c_{i,3}$, $c_{i,10}$, $c_{i,11}$ are all sets of π and one of $c_{i,6}$ or $c_{i,7}$ is a set of π. Assume first that $c_{i,6}$ is a set of π. Then, the subsets of the 2-components of VG_i that are also sets of π must be $c_{i,1} \setminus (c_{i,2} \cup c_{i,3})$, $c_{i,4} \setminus (c_{i,2} \cup c_{i,6})$, $c_{i,5} \setminus c_{i,3}$, $c_{i,7} \setminus (c_{i,5} \cup c_{i,6} \cup c_{i,9})$, $c_{i,8} \setminus (c_{i,6} \cup c_{i,10})$, $c_{i,9} \setminus c_{i,11}$, $c_{i,12} \setminus (c_{i,10} \cup c_{i,11})$. It is immediate to verify that the number of co-clustered pairs is 40. The case π contains set $c_{i,7}$ instead of $c_{i,6}$ is symmetric to the above one and thus lemma holds for every strict type (b) solution π for VG_i. □

Lemma 3.10. *Given Π, a vertex gadget VG_i, and a normal solution π, if the 2-components $c_{i,1}$ and $c_{i,12}$ are not both covered by π, then π is not optimal for Π_i.*

Proof. Let us recall that by Lemma 3.6 the set of 2-components covered by a normal solution π must correspond to an independent set of graph VG_i. We prove the lemma in the case π does not cover $c_{i,1}$, as indeed by construction of the vertex gadget this case is symmetric to the one Π does not contain $c_{i,12}$. In the following, first we show that we can construct from π a normal solution π^1 that covers $c_{i,1}, c_{i,4}, c_{i,5}$ and co-clusters more pairs than that of π. Then we show that we can construct a normal solution π' that is a type (a) solution (that is π' covers 2-components $\{c_{i,1}, c_{i,4}, c_{i,5}, c_{i,8}, c_{i,9}, c_{i,12}\}$) and co-clusters at least as many pairs as π^1.

Let us first show that we can construct from π a normal solution π^1 that covers $c_{i,1}$, $c_{i,4}$ and $c_{i,5}$. The following two cases must be considered.

Case (1). Assume that given triple $c_{i,2}$, $c_{i,4}$ and $c_{i,6}$, π covers both $c_{i,2}$ and $c_{i,6}$. Clearly, this fact implies that $c_{i,7}$ cannot be covered by π and thus either $c_{i,5}$ is just covered by π or otherwise by applying statement (2) of Lemma 3.7 we construct π^1 that covers $c_{i,5}$ and is of cost not worst than that of π.

Then, by applying statement (1) of Lemma 3.7 to triple $c_{i,1}$, $c_{i,3}$ and $c_{i,2}$ of π^1 we improve the solution in such a way that it covers $c_{i,1}$, but does not cover $c_{i,2}$ and $c_{i,4}$. Consequently, by applying statement (2) of Lemma 3.7 this time to triple $c_{i,2}$, $c_{i,4}$ and $c_{i,6}$, we obtain a normal solution π^1 that covers both $c_{i,4}$ and $c_{i,5}$ and which co-clusters more pairs than that of π, as required.

Case (2). Assume that given triple $c_{i,2}$, $c_{i,4}$ and $c_{i,6}$, π covers at most one of $c_{i,2}$ and $c_{i,6}$. Now, if given triple $c_{i,3}$, $c_{i,5}$, $c_{i,7}$, π covers both $c_{i,3}$ and $c_{i,7}$, symmetrically to case (1) we can built a normal solution π^1 which co-clusters more pairs than that of π and covers $c_{i,1}$, $c_{i,4}$ and $c_{i,5}$, as required. Thus assume that π covers at most one of $c_{i,3}$ and $c_{i,7}$. Thus, by applying statement (2) of Lemma 3.7 twice, first to triple $c_{i,2}$, $c_{i,4}$, $c_{i,6}$ and then to the triple $c_{i,3}$, $c_{i,5}$, $c_{i,7}$, we obtain a normal solution π^1 that covers both $c_{i,4}$ and $c_{i,5}$ and co-clusters at least as many pairs as π. Thus let us consider the normal solution π^1 that covers the 2-components $c_{i,4}$, $c_{i,5}$. Statement (1) of Lemma 3.7 can be applied to triple $c_{i,1}$, $c_{i,2}$ and $c_{i,3}$ so that π^1 is a normal solution that covers $c_{i,1}$, $c_{i,4}$, $c_{i,5}$ and which co-clusters more pairs than that of π, as required.

Now, having a normal solution π^1 that covers $c_{i,1}$, $c_{i,4}$ and $c_{i,5}$, since $c_{i,6}$ and $c_{i,7}$ are not covered by π^1, again by applying Lemma 3.7 to the two triples $c_{i,6}$, $c_{i,8}$, $c_{i,10}$ and $c_{i,7}$, $c_{i,9}$, $c_{i,11}$, we can obtain a solution π' so that it covers $c_{i,8}$ and $c_{i,9}$ and co-clusters at least as many pairs as π^1. Similarly, since $c_{i,10}$, $c_{i,11}$ are not covered by π', by applying Lemma 3.7 we can modify π' so that it also covers $c_{i,12}$ without decreasing the number of co-clustered pairs.

Since π' is a type (a) solution and it co-clusters more pairs than that of π, it follows that π contains at most 40 pairs. This fact concludes the proof. □

Lemma 3.11. *Given Π, a vertex gadget VG_i, and a normal solution π, if π is optimal for Π_i, then VG_i is a-induced in π.*

Proof. Lemma 3.10 shows that any optimal solution for Π_i must cover the 2-components $c_{i,1}$ and $c_{i,12}$. Repeatedly applying Lemma 3.7, it is easy to show that all optimal solution must cover also $c_{i,4}, c_{i,5}, c_{i,8}, c_{i,9}$, thus concluding the proof. □

Lemma 3.12. *Given Π, a normal solution π, and a vertex gadget VG_i, if π is not strict for VG_i then the number of pairs of $P(VG_i)$ co-clustered in π is at most* 40 *minus the number of docking vertices of VG_i for which π is not strict.*

Proof. The proof consists of modifying π so that the resulting solution π' is strict for VG_i, and analyzing the differences between the original and the final solution. For each non-strict docking vertex d of VG_i (with respect to π), move

its private elements to the set (possibly empty) $S_\pi(d)$. Let π' be the resulting solution. Since d is not a strict docking vertex, and π is a normal solution, in π at least one of the private elements belongs to a set that is not a 2-component of VG_i. Therefore $|S_{\pi'}(d)| > |S_\pi(d)|$. Notice that if v is a non-docking vertex of VG_i, then $S_{\pi'}(v) = S_\pi(v)$, hence the number of pairs of $P(VG_i)$ that are co-clustered in π' but not in π is at least as large as the number of non-strict docking vertices of VG_i in π.

Consequently if π' co-clusters at most 40 pairs of $P(VG_i)$, the lemma follows. By Lemmas 3.8, 3.11, if π' co-clusters more than 40 pairs of $P(VG_i)$, then VG_i is a-induced in π' which in turn implies that 41 pairs of $P(VG_i)$ are co-clustered in π' and $|S_{\pi'}(d)| = 4$ for all docking vertices d. Consequently the number of $P(VG_i)$ that are co-clustered in π' but not in π is at least three times the number of non-strict docking vertices of VG_i in π, hence the lemma follows. □

If π' in the proof of Lemma 3.12 is b-induced, then we are able to give a better measure of the number of pairs of $P(VG_i)$ co-clustered by π', as shown in the following corollary which follows from Lemma 3.9.

Corollary 3.13. *Given Π, a normal solution π, and a vertex gadget VG_i b-induced by π, then the number of pairs of $P(VG_i)$ co-clustered in π is exactly* 40 *minus the number of docking vertices of VG_i for which π is not strict.*

The main idea of the following part of the reduction consists of showing that we can restrict ourselves only to normal solutions π, where each vertex gadget VG_i is either a-induced or b-induced. This fact can be proved by modifying in polynomial time any normal solution π into a canonical solution π' without decreasing the number of co-clustered pairs.

Notice that the pairs assigned of $P(EG_{i,j})$ which can be co-clustered in normal solution (that is, pairs that are included in a 2-component of $EG_{i,j}$) are: $(j_3, h_{i,j,2})$, $(e_{i,j,1}, j_3)$, $(h_{i,j,2}, e_{i,j,1})$, $(j_4, e_{i,j,2})$, $(i_3, h_{i,j,1})$, $(e_{i,j,1}, i_3)$, $(h_{i,j,1}, e_{i,j,1})$, $(i_4, e_{i,j,2})$. Observe that, since $c_{i,j,1}$ and $c_{j,i,1}$ share an element, any solution π can cover at most one of $c_{i,j,1}$ and $c_{j,i,1}$. Similarly, π can cover at most one of $c_{i,j,2}$ and $c_{j,i,2}$ and at most five pairs of $P(EG_{i,j})$ are co-clustered in π.

Lemma 3.14. *Given Π, two adjacent vertex gadgets VG_i and VG_j and a normal solution π that is strict for both docking vertices $d_i(EG_{i,j})$, $d_j(EG_{i,j})$, then π co-clusters at most one pair of $P(EG_{i,j})$.*

Proof. Since $d_i(EG_{i,j})$, $d_j(EG_{i,j})$ are strict, their private elements belongs to $S_\pi(d_i(EG_{i,j}))$, $S_\pi(d_j(EG_{i,j}))$, therefore the only co-clustered pairs of $P(EG_{i,j})$ co-clustered in π might be $(h_{i,j,2}, e_{i,j,1})$ and $(h_{i,j,1}, e_{i,j,1})$. Since those pairs share element $e_{i,j,1}$ and are not subset of a same 2-component, only one of those pairs can be co-clustered in a normal solution. □

Lemma 3.15. *Given Π, two adjacent vertex gadgets VG_i and VG_j and a normal solution π that co-clusters* 5 *pairs of $P(EG_{i,j})$, then π is not strict for neither $d_i(EG_{i,j})$, nor $d_j(EG_{i,j})$.*

Proof. By Lemma 3.14, at most one of $d_i(EG_{i,j})$ and $d_j(EG_{i,j})$ is strict. Assume that $d_j(EG_{j,i})$ is strict, then the only pairs of $P(EG_{i,j})$ that might be co-clustered in π are $(h_{i,j,2}, e_{i,j,1})$, $(i_3, h_{i,j,1})$, $(e_{i,j,1}, i_3)$, $(h_{i,j,1}, e_{i,j,1})$, $(i_4, e_{i,j,2})$. Since $(h_{i,j,2}, e_{i,j,1})$ and $(h_{i,j,1}, e_{i,j,1})$ share element $e_{i,j,1}$ and are not subset of a same 2-component, only one of these can be co-clustered in a normal solution, contradicting the assumption that π co-clusters 5 pairs of $P(EG_{i,j})$. □

Lemma 3.16. *Given Π, two adjacent vertex gadgets VG_i and VG_j and a normal solution π, it is possible to compute in polynomial time a normal solution π' so that*:

(1) *π' is of type* (b) *or type* (a) *for both vertices VG_i and VG_j and is of type* (b) *for at least one vertex of VG_i and VG_j,*
(2) *π' co-clusters exactly four pairs of $P(EG_{i,j})$,*
(3) *π' is strict for exactly one of the docking vertices of $EG_{i,j}$,*
(4) *π' co-clusters at least as many pairs as π.*

Proof. We will construct the solution π' from π as follows. Let VG_i be a vertex gadget. If π is strict for VG_i, we define a type (a) solution for VG_i in π'. By Lemma 3.11 a type (a) solution co-clusters the maximum number of pairs of $P(VG_i)$, therefore π' co-clusters at least as many pairs of $P(VG_i)$ as π.

If π is not strict for VG_i, by Lemma 3.12 the number of pairs of $P(VG_i)$ co-clustered by π is at most 40 minus the number of docking vertices of VG_i that are not strict in π.

Define a type (b) solution for VG_i in π' so that a docking vertex of VG_i is strict in π' iff is strict in π. Since by Corollary 3.13 the number of pairs of $P(VG_i)$ co-clustered by a type (b) solution is 40 minus the number of docking vertices of VG_i, then π' co-clusters at least as many pairs of $P(VG_i)$ as π.

Notice that by construction, each vertex gadget is either a-induced or b-induced in π' as required in statement (1) of the lemma. Observe that, for each edge gadget $EG_{i,j}$, the pairs of $P(EG_{i,j})$ are unaffected by construction of π'. Let us now modify π' so that it is not strict type (b) for exactly one of the two vertices VG_i and VG_j and is strict for exactly one of the docking vertices of $EG_{i,j}$. Hence statements (2) and (3) of the lemma hold, while π' still co-clusters at least as many pairs as π.

Now consider an edge gadget $EG_{i,j}$, we will consider three cases depending on the number of strict docking vertices of $EG_{i,j}$.

Assume first that both docking vertices $d_i(EG_{i,j})$, $d_j(EG_{i,j})$ are strict, then we have to distinguish two subcases. Recall that $d_i(EG_{i,j})$ is the docking vertex shared by VG_i and $EG_{i,j}$. If one of VG_i, VG_j is b-induced in π', assume w.l.o.g. that VG_i is b-induced. In π' make the 2-components $c_{i,j,1}$ and $c_{i,j,2}$ covered, eventually moving the elements of $c_{i,j,1}$ and $c_{i,j,2}$ from their sets in π'. Notice that π' co-clusters four pairs of $P(EG_{i,j})$, while the number of strict vertices of VG_i in π' is equal to that in π minus one, as $d_i(EG_{i,j})$ is no more strict in π'. Applying Corollary 3.13 we obtain that π' co-clusters more pairs than π, while noticing that $d_j(EG_{i,j})$ is unaffected we obtain that $d_j(EG_{i,j})$ is the only strict docking vertex of $EG_{i,j}$.

The second subcase holds when both VG_i, VG_j are a-induced in π'. Modify π' so that VG_i is strict b-induced. By applying Lemmas 3.8 and 3.9 the number of co-clustered pairs of $P(VG_i)$ is decreased by one. Now VG_i is b-induced and we can follow the same argument of the first subcase. The overall effect is that the number of co-clustered pairs is not decreased, completing the analysis of the case where both docking vertices are strict.

Assume now that no docking vertex of $EG_{i,j}$ is strict. By construction both VG_i and VG_j are b-induced. Modify π' so that $c_{i,j,1}$ and $c_{i,j,2}$ are covered and the docking vertex $d_j(EG_{i,j})$ shared between VG_j and $EG_{i,j}$ is strict. Notice that now four pairs of $P(EG_{i,j})$ are co-clustered in π'. Indeed by Lemma 3.15, since at least one docking vertex of $EG_{i,j}$ is strict, at most four pairs of $P(EG_{i,j})$ are co-clustered in π', while previously at most five pairs of $P(EG_{i,j})$ were co-clustered. At the same time the number of strict docking vertices of VG_j is increased by one. By Corollary 3.13 the overall effect is that the number of co-clustered pairs in π' is at least as large as that in π.

W.l.o.g. the last case we have to consider is when $d_j(EG_{i,j})$ is strict but $d_i(EG_{i,j})$ is not. In such case modify π' so that $c_{i,j,1}$ and $c_{i,j,2}$ are covered. Notice that now four pairs of $P(EG_{i,j})$ are co-clustered. By Lemma 3.15, at most four pairs of $P(EG_{i,j})$ are co-clustered in π, completing the proof. □

Recall that a normal solution satisfying Lemma 3.16 is called *canonical*.

Corollary 3.17. *Let π be a canonical solution for Π. Then no two a-induced vertex gadgets are adjacent.*

The following corollary, which establishes the number of co-clustered pairs in a canonical solution, is an immediate consequence of Lemmas 3.8 and 3.16, Corollary 3.13 and the observation that the number of non-strict docking vertices is equal to m.

Corollary 3.18. *Let π be a canonical solution for Π, such that h vertex gadgets are a-induced and $n-h$ vertex gadgets are b-induced by π. Then π co-clusters $41h + 40(n-h) + 3m$ pairs.*

Given a normal solution π^* with at least $41h + 40(n-h) + 3m$ pairs, by Corollary 3.17 it is easy to find in polynomial time a set of h independent vertex gadgets in the G-graph. The proof of the L-reduction is completed by noting that the cost of a solution is strictly related by a constant to the number of pairs in the solution and parameters n, h, where n is related to h by a constant (n is the number of vertices in a cubic graph).

Theorem 3.19. *There is an independent set I of a G-graph $\mathcal{G} = (V, E)$ of size $6k + 5(n-k) + 2m$ if and only if there is a solution π of the associated instance of* MR3CC *with $41k + 40(n-k) + 3m$ co-clustered pairs, where n, m and k are respectively the number of vertex gadgets, the number of edge gadgets and the number of mutually independent vertex gadgets.*

Proof. Let I be an independent set of $\mathcal{G}$ with cardinality $6k + 5(n-k) + 2m$. By construction of $\mathcal{G}$, there are k vertex gadgets with an independent set of type 1 and $n-k$ vertex gadgets with an independent set of type 2. Since an edge gadget does not connect two vertex gadgets with an independent set of type 1, there exists a canonical solution π^* of MR3CC having a type (a) solution for vertex gadgets with an independent set of type 1, a type (b) solution for vertex gadgets with an independent set of type 2 and covering two 2-components associated with each edge gadget. By Corollary 3.18 the number of co-clustered pairs in π^* is $41k + 40(n-k) + 3m$.

Now consider a solution π of MR3CC with $41k + 40(n-k) + 3m$ co-clustered pairs. By Lemma 3.16 we can assume that π is a canonical solution. Moreover by Corollary 3.18, there are k vertex gadgets of $\mathcal{G}$ that are a-induced in π. Since π is canonical such a-induced vertex gadgets are mutually non-adjacent, by Corollary 3.17. We can construct an independent set I of $\mathcal{G}$ simply assigning a *type* 1 independent set to each vertex gadget that is a-induced in π', assigning a *type* 2 independent set to each vertex gadget that is b-induced in π', and assigning two vertices for each edge gadget.

Since the set of 2-components covered by π' must form an independent set of $\mathcal{G}$ by Lemma 3.6, we have built an independent set consisting of $6k + 5(n-k) + 2m$ covered 2-components. □

4. The PTAS

In this section, we will present two closely related polynomial-time approximation schemes, one for the MAXIMUM CORRELATION CLUSTERING problem when the ratio between the maximum and the minimum weights is upper bounded by a constant, while the second PTAS is for the MAXIMUM CONSENSUS CLUSTERING problem. Both algorithms are based on the smooth polynomial programming technique of [3], but they are not straightforward applications of the technique.

We will briefly recall the relevant material from this paper, rephrased to take into account maximization problems instead of minimization. A *c-smooth polynomial integer program* (or PIP) over variables $x_1, \ldots, x_m$ is a problem of the form:

$$\begin{aligned} &\text{maximize} && p_0(x_1, \ldots, x_m), \\ &\text{subject to} && l_j \leqslant p_j(x_1, \ldots, x_m) \leqslant u_j, \\ & && x_i \in \{0, 1\} \quad \text{for } 1 \leqslant i \leqslant m, \end{aligned} \tag{3}$$

where each p_j is an m-variate polynomial of maximum degree d, and coefficients of each degree-ℓ monomial (term) are in the interval $[-cm^{d-\ell}, cm^{d-\ell}]$. Let *opt* denote the optimal value of a PIP. The fundamental result that we will use, Theorem 1.10 of [3], asserts that, for each $\delta_1 > 0$, there exists an approximation algorithm that, in time $O(m^{\delta_1^{-2}})$, computes a 0/1 assignment $\langle y_1, \ldots, y_m \rangle$ to the variables $\langle x_1, \ldots, x_m \rangle$ of a c-smooth PIP whose value is at least $opt - \delta_1 m^d$. Moreover the assignment $\langle y_1, \ldots, y_m \rangle$ satisfies each linear constraint within an *additive* error of $O(\delta_1 \sqrt{n \log n})$. Notice that $\langle y_1, \ldots, y_m \rangle$ is not necessarily a feasible solution, therefore applying Theorem 1.10 of [3] as a black box in not sufficient to describe an approximation algorithm, since we have to ensure that the solution computed is feasible.

We are now ready for describing our algorithm for MAXIMUM CORRELATION CLUSTERING recalling that the value of a partition π is

$$\sum_{e=(i,j)} \big(a(i,j) r_\pi(i,j) + b(i,j)\big(1 - r_\pi(i,j)\big)\big),$$

where $r_\pi(i,j) = 1$ if and only if the elements i and j are co-clustered in π and $r_\pi(i,j) = 0$ otherwise. Since we are interested only in instances where the ratio $\max_{i,j}\{a(i,j), b(i,j)\} / \min_{i,j}\{a(i,j), b(i,j)\}$ is at most a constant, it is not restrictive to assume that $\min_{i,j}\{a(i,j), b(i,j)\} = 1$ and $\max_{i,j}\{a(i,j), b(i,j)\}$ is equal to a constant $w_{\max}$.

It has already been shown in [6] that the optimal value of MAXIMUM CORRELATION CLUSTERING is $\Theta(n^2)$ (where n is the number of elements in V). By this fact, it follows that an approximation algorithm with additive error δn^2 is a $(1+\frac{\delta}{c})$ approximation algorithm for a fixed constant c, which proves the existence of a PTAS for the problem.

Notice that Theorem 1.10 of [3] guarantees a $\delta_1 m^d$ additional error, therefore there are two possibilities: either we describe MAXIMUM CORRELATION CLUSTERING as a linear PIP with $m = O(n^2)$ variables, or as a quadratic PIP with $m = O(n)$ variables. We will follow the latter route. The simplest formulation uses variables $x_{i,j}$ whose value is 1 if and only if the ith element is in the jth set of the partition, consequently the quantity $\sum_t x_{i,t}x_{j,t}$ is equal to 1 if and only if the ith and the jth element are in the same set of the partition, otherwise it is equal to 0. The constraints in our PIP must enforce that any 0/1 assignment to the variables leads to a partition, that is each element belongs to exactly one set of the partition. The formulation follows:

$$\begin{aligned} &\text{maximize} && \sum_{i,j}\left(a_{i,j}\sum_t x_{i,t}x_{j,t} + b_{i,j}\left(1-\sum_t x_{i,t}x_{j,t}\right)\right),\\ &\text{subject to} && \sum_t x_{i,t} = 1 \quad \text{for } 1 \leqslant i \leqslant n,\\ &&& x_{i,t} \in \{0,1\}. \end{aligned} \tag{4}$$

A step of the algorithm described in [3] consists of setting each variable $x_{i,t}$ to 0 or 1 independently and randomly with probability $\tilde{x}_{i,t}$, where each $\tilde{x}_{i,t}$ is computed by optimally solving a certain linear program. Unfortunately this fact does not guarantee that the resulting 0/1 solution is feasible because each constraint might be violated, albeit by a small quantity. Exploiting the structure of the constraints can lead to guaranteeing the feasibility of the 0/1 solution; such approach has been pioneered in [4] and refined in [12] for a set of linear constraints very similar to ours. Here we follow the idea of the latter paper, by choosing the values of $x_{i,j}$ in a dependent way; more precisely with probability $\tilde{x}_{i,t}$, all variables $x_{i,j}$, except for $x_{i,t}$, are set to 0, and the variable $x_{i,t}$ is set to 1. With this modified randomized rounding algorithm it is possible to prove that Theorem 1.10 of [3] holds and the resulting 0/1 solution actually encodes a partition.

Now that we have guaranteed that the solution is a partition, we can concentrate on the error produced by the algorithm. Since the above PIP is quadratic, in order to have an additive error $O(n^2)$, it is necessary to have $O(n)$ variables. Unfortunately there may be $\Theta(n)$ sets in the partition and therefore there may be $\Theta(n^2)$ variables of the form $x_{i,j}$, therefore it is necessary to formulate a similar PIP with $O(n)$ variables. The next step is to show that considering only partitions with at most $8w_{\max}/\delta_2$ sets suffices, as the optimum over partitions with at most $8w_{\max}/\delta_2$ sets is within an additive error $\delta_2 n^2$ of the optimal unrestricted partition.

Lemma 4.1. *Let Π be an optimal solution of an instance I of* MAXIMUM CORRELATION CLUSTERING *and let $\delta_2 > 0$ be an arbitrary constant. Then there exists a solution Π_1 of I with at most $8w_{\max}/\delta_2$ (i.e. a constant) number of sets, and such that the values of Π and of Π_1 differ by at most $\delta_2 n^2$.*

Proof. Notice that the quantity $\delta_2/4w_{\max}$ is a constant. Classify the sets in Π into large sets (containing more than $\delta_2 n/4w_{\max}$ elements) and small sets (containing at most $\delta_2 n/4w_{\max}$ elements). It is immediate to note that merging two small sets gives a set of at most $\delta_2 n/2w_{\max}$ elements. Initially let $\Pi_1 = \Pi$. Repeatedly merge two small sets in Π_1 until at most one small set remains in Π_1. When the above procedure ends, in Π_1 there is at most one small set. Remember that large sets have at least $\delta_2 n/4w_{\max}$ elements, therefore in Π_1 there are at most $4w_{\max}/\delta_2$ large sets.

Now we can show that the value of Π is close to that of Π_1. Since all sets in Π_1 are union of sets in Π all co-clustered pairs in Π are also co-clustered in Π_1, therefore the only pairs that can be clustered differently in Π and in Π_1 are pairs entirely contained in a set in $\Pi_1 - \Pi$. Let X be a set $\Pi_1 - \Pi$: in the worst case all pairs of elements in X are clustered differently in Π_1 and Π. By construction of Π_1, $|X| \leqslant \delta_2 n/2w_{\max}$, therefore the number of pairs contained in X are at most $\delta_2^2 n^2/8w_{\max}^2$, each one with maximum weight $w_{\max}$. Since there are at most $4w_{\max}/\delta_2 + 1 \leqslant 8w_{\max}/\delta_2$ sets in $\Pi_1 - \Pi$, the difference of the values of Π and Π_1 is at most $w_{\max}(\delta_2^2 n^2/8w_{\max}^2)(8w_{\max}/\delta_2) = \delta_2 n^2$. □

Clearly the PIP in Eq. (4) is also a formulation for the restriction of MAXIMUM CORRELATION CLUSTERING where the number of sets in the computed partition is at most $8w_{\max}/\delta_2$; in fact it suffices to have only the variables

$x_{i,j}$ with $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant \lceil 8w_{\max}/\delta_2 \rceil$. Therefore we can apply Theorem 1.10 of [3] to obtain a 0/1 assignment to the variables $x_{i,j}$ with additional error at most $\delta_2 n^2$ for such restriction. By Lemma 4.1 we know that the optimum of the restricted problem is within an additive error of $\delta_1 n^2$ of the optimum of MAXIMUM CORRELATION CLUSTERING. Summing up the two additional errors, and setting $\delta_1 + \delta_2 = \epsilon$ leads to an overall additional error ϵn^2 for MAXIMUM CORRELATION CLUSTERING.

The PTAS similar to the one described above can be applied also to the problem of MAXIMUM CONSENSUS CLUSTERING where we are given a set $\{\pi_1, \pi_2, \ldots, \pi_k\}$ of partitions over universe U, where k is unbounded and part of the input, and we want to find a partition π of the elements of U maximizing $\sum_{i=1}^{k} s(\pi_i, \pi)$, where $s(\cdot,\cdot)$ is the similarity value of two partitions. The problem is a restriction of MAXIMUM CORRELATION CLUSTERING, where the coefficient $a(i, j)$ is the number of input partitions where i and j are co-clustered, and $b(i, j) = k - a(i, j)$. Unfortunately the PTAS for MAXIMUM CORRELATION CLUSTERING cannot be applied directly, as the ratio between the maximum and the minimum of the coefficients $a_{i,j}$, $b_{i,j}$ may not be upper bounded by any constant, e.g. when a pair is co-clustered in no input partition the minimum $a_{i,j}$ is zero; in such case the PIP in (4) is not c-smooth for any constant c.

Notice that the optimal solution value for the problem is $\Theta(kn^2)$. To see this, let us consider two solutions, C_1 and C_2, where C_1 corresponds to a partition consisting of only one set and C_2 consists of the partition of the universe set into singletons. The sum of the values of C_1 and C_2 is exactly $k\binom{n}{2}$, and therefore the best of C_1 and C_2 has value at least $\frac{k}{2}\binom{n}{2}$. Consequently finding a feasible solution with additive error at most ϵkn^2 suffices to obtain a PTAS.

The first step is to discretize the coefficients by rounding them to a multiple of k/d (for some constant d that will be determined later). More specifically, we pose $a'_{i,j} = \lfloor a_{i,j} d/k \rfloor \cdot k/d$ and $b'_{i,j} = k - a'_{i,j}$. Now we must estimate the additional error introduced when replacing the coefficients $a_{i,j}$ and $b_{i,j}$ with $a'_{i,j}$, $b'_{i,j}$. It is immediate to notice that $|a_{i,j} - a'_{i,j}| \leqslant k/d$. Therefore, a crude estimate of the overall error gives at most $\frac{1}{d} kn^2$.

The resulting PIP is not c-smooth for any constant c, as the coefficients of the objective function are unbounded. Hence, we need to divide all the coefficients by k/d, and set $a''_{i,j} = a'_{i,j} d/k$ and $b''_{i,j} = b'_{i,j} d/k$. Notice that all the coefficients $a''_{i,j}$ and $b''_{i,j}$ belong to the set $\{0, 1, \ldots, d\}$. Now we can present our PIP:

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j} \left(a''_{i,j} \sum_t x_{i,t} x_{j,t} + b''_{i,j} \left(1 - \sum_t x_{i,t} x_{j,t} \right) \right), \\ \text{subject to} \quad & \sum_t x_{i,t} = 1 \quad \text{for } 1 \leqslant i \leqslant n, \\ & x_{i,t} \in \{0, 1\}. \end{aligned} \tag{5}$$

Just as in the case of MAXIMUM CORRELATION CLUSTERING, we have to limit the number of sets in the output partition to be at most $8d/\delta_2$ (d is an upper bound of $\max_{i,j}\{a''_{i,j}, b''_{i,j}\}$). Analogously as in Lemma 4.1, this restriction introduces an additive error not larger than $\delta_2 n^2$. Therefore, applying Theorem 1.10 of [3] with the modified randomized rounding to obtain a 0/1 assignment to the variables $x_{i,j}$, we obtain a solution with additive error $(\delta_1 + \delta_2)n^2$. Re-scaling back to the coefficients $a'_{i,j}$, $b'_{i,j}$ yields a solution for the MAXIMUM CONSENSUS CLUSTERING problem on coefficients $a'_{i,j}$ and $b'_{i,j}$ with an additive error at most $\frac{(\delta_1+\delta_2)}{d} kn^2$.

Adding the above error to the one introduced by rounding the coefficients $a_{i,j}$, $b_{i,j}$ to $a'_{i,j}$, $b'_{i,j}$ gives an overall additive error $\frac{(\delta_1+\delta_2)}{d} kn^2 + \frac{1}{d} kn^2 = \frac{(\delta_1+\delta_2+1)}{d} kn^2$. Since we want an additive error at most ϵkn^2 and δ_1, δ_2 are two arbitrarily small constants, setting $d = \frac{3}{\epsilon}$ leads to an overall additive error of $\frac{(\delta_1+\delta_2+1)}{3} \epsilon kn^2 \leqslant \epsilon kn^2$. This completes the construction of the PTAS.

5. A $\frac{4}{5}$-approximation algorithm for max Consensus Clustering

In this section, we present a combinatorial $\frac{4}{5}$-approximation algorithm for MAX CONSENSUS CLUSTERING on instances of 3 partitions (MAX-3CC). Note that for MIN CONSENSUS CLUSTERING on instances of 3 partitions, the trivial approximation algorithm that picks the input partition that minimizes the symmetric difference distance has approximation factor $\frac{3}{4}$. Indeed, the symmetric difference distance $d(\pi_1, \pi_2)$ is a metric which immediately leads to

a $2-\frac{2}{k}$-approximation via the center-star technique (see [15] for an application of the technique). An instance of MAX-3CC consists of a set of partitions $\Pi = \{\pi_1, \pi_2, \pi_3\}$ over universe U. In what follows, given a set X, we use $P(X)$ to denote the set of all pairs of elements over X. The algorithm constructs a partition by selecting 2-components (i.e. co-clustered pairs), using a greedy technique, from a component graph built from the input instance.

Algorithm 3. GREEDY-REDUCED-CC

1. **Input:** an instance Π over universe U
2. $\pi \leftarrow \emptyset$
3. $\Pi' \leftarrow \Pi$
4. $G' \leftarrow$ the component graph associated with Π'
5. **While** $|V(G')| > 0$ **do**
6. X is a maximum cardinality 2-component in G'
7. $\pi \leftarrow \pi \cup \{X\}$
8. remove all elements of X from U and update Π'
9. $G' \leftarrow$ the component graph associated with Π'
10. **For each** $u \in U$, u is not in a set of π **do**
11. add $\{u\}$ to π
12. **Return** π

Let $C_1, \ldots, C_n$ be the 2-components of the instance. Denote with $\mathcal{A} = \{A_1, \ldots, A_k\}$ the set of non-empty intersections of pairs of 2-components, that is $\mathcal{A} = \{C_i \cap C_j\colon\ C_i \cap C_j \neq \emptyset\}$. By Lemma 2.3, sets in $\mathcal{A}$ are pairwise disjoint. Note that, if $|A_i| \geqslant 2$, then each pair of elements $(x, y) \in A_i$ is co-clustered in all three input partitions.

Let Π be an instance of MAX-3CC and let (i, j) be a pair of elements of U. Recall that $s_\Pi(i, j)$ denotes the number of input partitions in which i and j are co-clustered, while $d_\Pi(i, j)$ denotes the number of input partitions in which i and j are not co-clustered. Let $w_{\mathrm{MAX},\Pi}(i, j)$ denote the maximum of $s_\Pi(i, j)$ and $d_\Pi(i, j)$. Given a solution π, we define the weight of (i, j) in π, denoted by $w_\pi(i, j)$, as $s_\Pi(i, j)$ if (i, j) is co-clustered in π, $d_\Pi(i, j)$ otherwise. Thus the similarity value $v(\pi)$ of a solution π can be expressed simply as $\sum_{i<j} w_\pi(i, j)$. Let *opt* be an optimal solution of Π. Then for each $p \in P(U)$, $w_{\mathrm{opt}}(p) \leqslant w_{\mathrm{MAX},\Pi}(p)$. Let P' be a set of pairs, the *similarity value* of P' in a solution π is $v(P', \pi) = \sum_{p \in P'} w_\pi(p)$.

Let π be the solution returned by Algorithm 3. Since every set in the solution π is a subset of a 2-component, then all elements co-clustered in π are co-clustered in at least two partitions of the input and thus $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p)$ for each pair p of elements co-clustered in a set of solution π.

Moreover, observe that, since GREEDY-REDUCED-CC constructs a partition π from subsets of 2-components, if a pair p is co-clustered in the instance in less than two partitions, then p is not co-clustered in π and thus $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p)$. Similarly, it is easy to verify that $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p) = 3$ for each pair p co-clustered in the instance in three partitions, since p is co-clustered in π. Let $P_{\pi,3\mathrm{IN}}$ denote the set of such pairs.

Consequently, $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p)$ except for some pairs $p = (a, b)$, such that $a, b \in X$, where X is a 2-component of graph G and a, b are not co-clustered in π. Indeed, in this case $w_\pi(p) = 1 < w_{\mathrm{MAX},\Pi}(p) = 2$. Let us denote by $P_{\pi,\mathrm{loss}}$ the set of such pairs. Observe that by construction of GREEDY-REDUCED-CC, given a 2-component Z such that $A_i \subseteq Z$, a pair $p = (x, y) \in P_{\pi,\mathrm{loss}}$ consists of an element $x \in A_i$ and an element y that belongs to $Z \setminus A_i$. We will show that the number of pairs in $P_{\pi,\mathrm{loss}}$ is limited and we can bound from above the similarity value of such pairs as stated in Lemma 5.1.

Let us denote by $P_{\pi,\mathrm{IN}}$ the set of pairs contained in 2-components of G such that $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p) = 2$, by $P_{\pi,\mathrm{OUT}}$ the subset of pairs p that are not in 2-components and such that $w_\pi(p) = w_{\mathrm{MAX},\Pi}(p) = 2$ (p is not co-clustered in π). Moreover, denote by P_{A_i} the set of pairs with elements in a certain A_i (where $A_i \in \mathcal{A}$) and $P_{\mathcal{A}} = \bigcup_i P_{A_i}$. Let $v(\mathcal{A})$ denote the similarity value of the pairs in P_A, that is $v(\mathcal{A}) = v(P_A, \pi) = \sum_{A_i \in \mathcal{A}} \frac{3|A_i|(|A_i|-1)}{2}$, since the sets in $\mathcal{A}$ are pairwise disjoint by Lemma 2.3.

Let π be a solution returned by GREEDY-REDUCED-CC and *OPT* the optimal solution. Observe that $P_{\pi,\mathrm{IN}} \cup P_A$ is the set of pairs co-clustered in π, while the set of all pairs not co-clustered in π consists of $P_{\pi,\mathrm{OUT}} \cup P_{\pi,\mathrm{loss}} \cup P_{\pi,3\mathrm{OUT}}$. Consequently, $v(\pi) = v(P_{\pi,\mathrm{IN}}, \pi) + v(\mathcal{A}) + v(P_{\pi,\mathrm{OUT}}, \pi) + v(P_{\pi,\mathrm{loss}}, \pi) + v(P_{\pi,3\mathrm{OUT}}, \pi)$.

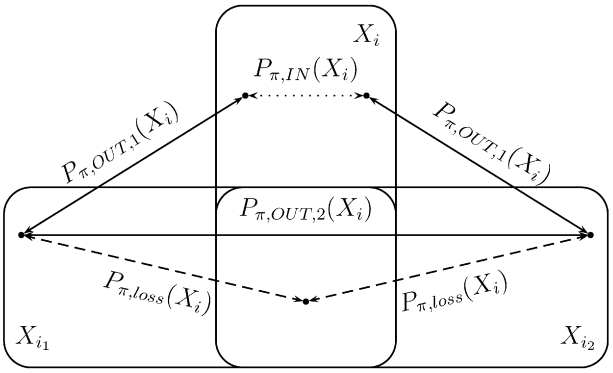


Fig. 6. The 2-component X_i and the set of pairs $P_{\pi,\mathrm{IN}}(X_i)$, $P_{\pi,\mathrm{OUT},1}(X_i)$, $P_{\pi,\mathrm{OUT},2}(X_i)$, $P_{\pi,\mathrm{loss}}(X_i)$.

Moreover, $v(OPT) \leqslant \sum_p w_{\mathrm{MAX},\Pi}(p) = v(P_{\pi,\mathrm{IN}}, \pi) + v(\mathcal{A}) + v(P_{\pi,\mathrm{OUT}}, \pi) + 2v(P_{\pi,\mathrm{loss}}, \pi) + v(P_{\pi,3\mathrm{OUT}}, \pi) = v(\pi) + v(P_{\pi,\mathrm{loss}}, \pi)$.

The following basic result is used to prove the approximation factor in Theorem 5.2.

Lemma 5.1. *Let π be a solution computed by Algorithm* 3*, then the following holds*:

$$v(P_{\pi,\mathrm{IN}}, \pi) + v(\mathcal{A}) + v(P_{\pi,\mathrm{OUT}}, \pi) \geqslant 3v(P_{\pi,\mathrm{loss}}, \pi).$$

Theorem 5.2. *Algorithm* GREEDY-REDUCED-CC *achieves an approximation factor of* $\frac{4}{5}$.

Proof. As observed before, $v(OPT) \leqslant v(\pi) + v(P_{\pi,\mathrm{loss}}, \pi)$. Adding $v(P_{\pi,\mathrm{loss}}, \pi)$ to both sides of inequality in Lemma 5.1, we obtain $4v(P_{\pi,\mathrm{loss}}, \pi) \leqslant v(\pi)$, which proves that $v(OPT) \leqslant \frac{5}{4}v(\pi)$. □

Let $\langle X_1, X_2, \ldots, X_z \rangle$ be the sequence of sets of size at least 2 added to the solution π by GREEDY-REDUCED-CC at successive steps $1, 2, \ldots, z$. Thus let U_i denote the universe set of the component graph G_i obtained when component X_i is chosen by GREEDY-REDUCED-CC. Observe that we will prove the inequality of Lemma 5.1 by showing in Lemma 5.6 that a similar inequality holds for the pairs considered at each iteration i of the algorithm GREEDY-REDUCED-CC; such pairs are over universe U_i and are defined by means of the 2-component graph G_i and set X_i as follows.

Denote by $P_{\pi,\mathrm{IN}}(X_i)$ the set of the pairs in $P_{\pi,\mathrm{IN}}$ having two elements in X_i. Moreover, denote by $P_{\pi,\mathrm{loss}}(X_i)$ the set of pairs in $P_{\pi,\mathrm{loss}}$ having one element in X_i and consisting of elements over universe U_i. Given X_i, observe that by Lemma 2.3 there are at most only 2-components of graph G_i, denoted as X_{i_1}, X_{i_2}, sharing a common element of X_i and moreover $X_{i_1} \cap X_{i_2} = X_i \cap X_{i_1} = X_i \cap X_{i_2}$. Thus, denote by $P_{\pi,\mathrm{OUT},1}(X_i)$ the set of pairs $(a, b) \in P_{\pi,\mathrm{OUT}}$ with $a, b \in U_i$, $a \in X_i$ and $b \in X_{i_1} \cup X_{i_2}$. Denote by $P_{\pi,\mathrm{OUT},2}(X_i)$ the set of pairs $(a, b) \in P_{\pi,\mathrm{OUT}}$ with $a, b \in U_i$, such that a, b belong to X_{i_1}, X_{i_2} respectively. Figure 6 illustrates sets of pairs defined above.

By construction, $P_{\pi,\mathrm{IN}}(X_i)$, $P_{\pi,\mathrm{loss}}(X_i)$, $P_{\pi,\mathrm{OUT},1}(X_i)$, $P_{\pi,\mathrm{OUT},2}(X_i)$ are all pairwise disjoint sets. Indeed, pairs in $P_{\pi,\mathrm{IN}}(X_i)$ are contained in X_i, while pairs in $P_{\pi,\mathrm{loss}}(X_i)$ and $P_{\pi,\mathrm{OUT},1}(X_i)$ have exactly one element in X_i and pairs in $P_{\pi,\mathrm{OUT},2}(X_i)$ have no element in X_i. Moreover, pairs in $P_{\pi,\mathrm{loss}}$ are in 2-components of graph G, differently from pairs in $P_{\pi,\mathrm{OUT}}$.

A fundamental step in proving Lemma 5.1 is the following claim:

Claim 5.3. *Let X_i and X_j be sets added to the solution π by* GREEDY-REDUCED-CC *at two distinct steps i, j. Then $P_{\pi,\mathrm{IN}}(X_i)$, $P_{\pi,\mathrm{IN}}(X_j)$, $P_{\pi,\mathrm{OUT},1}(X_i)$, $P_{\pi,\mathrm{OUT},2}(X_i)$, $P_{\pi,\mathrm{OUT},1}(X_j)$, $P_{\pi,\mathrm{loss}}(X_i)$ and $P_{\pi,\mathrm{loss}}(X_j)$ are all pairwise disjoint sets.*

Proof. Assume w.l.o.g. that $i < j$, that is X_i is added to the solution π before set X_j. Since each element in X_i is deleted from U_i after adding X_i to π, it follows that all pairs with an element in X_i, that is $P_{\pi,\mathrm{IN}}(X_i)$, $P_{\pi,\mathrm{OUT},1}(X_i)$, $P_{\pi,\mathrm{loss}}(X_i)$, are pairwise disjoint from pairs in $P_{\pi,\mathrm{IN}}(X_j)$, $P_{\pi,\mathrm{loss}}(X_j)$, $P_{\pi,\mathrm{OUT},1}(X_j)$ consisting of elements distinct

from those in X_i. Moreover, $P_{\pi,\mathrm{OUT},2}(X_i)$ consists of pairs that are not in any 2-component and are disjoint from pairs in $P_{\pi,\mathrm{OUT},1}(X_j)$, since by Lemma 2.3, there is no 2-component (including X_j) different from X_i sharing elements with both X_{i1}, X_{i2}. Since the other pairs in $P_{\pi,\mathrm{IN}}(X_i)$, $P_{\pi,\mathrm{IN}}(X_j)$, $P_{\pi,\mathrm{loss}}(X_i)$ and $P_{\pi,\mathrm{loss}}(X_j)$ are all contained in 2-components, this fact concludes the proof of the lemma. □

By the above Claim 5.3, it follows that $P_{\pi,\mathrm{OUT},2}(X_i)$ might share pairs only with a set $P_{\pi,\mathrm{OUT},2}(X_j)$. In what follows we will show that there exists at most one set X_j such that $(a,b) \in P_{\pi,\mathrm{OUT},2}(X_i)$ and $(a,b) \in P_{\pi,\mathrm{OUT},2}(X_j)$ (see Claim 5.5).

Lemma 5.4. *Let X_i and X_j be two sets added to the solution π by* GREEDY-REDUCED-CC *at two distinct steps i, j, with $i < j$. Let X_{i_1}, X_{i_2} be two 2-components such that $X_i \cap X_{i_1} = X_i \cap X_{i_2} = X_{i_1} \cap X_{i_2} \neq \emptyset$ and let X_{j_1}, X_{j_2} be two 2-components such that $X_j \cap X_{j_1} = X_j \cap X_{j_2} = X_{j_1} \cap X_{j_2} \neq \emptyset$. If $X_{i_1} \cap X_{j_1} = A \neq \emptyset$ and $X_{i_2} \cap X_{j_2} = B \neq \emptyset$, then each 2-component different from X_{i_1}, X_{j_1} that contains elements of A shares no element with any 2-component different from X_{i_2}, X_{j_2} that contains elements of B.*

Proof. By Corollary 2.4, w.l.o.g. we can assume that $X_i \cup X_{i_1}$ is included in a set Z_1 of π_1, $X_i \cup X_{i_2}$ is included in a set Z_2 of π_2, $X_{i_1} \cup X_{i_2}$ is included in a set Z_3 of π_3. Furthermore observe that, by maximality of 2-components, π_1 must not contain X_{i_2}, π_2 must not contain X_{i_1} and π_3 must not contain X_i.

A similar property must hold for X_j, where by construction of the algorithm X_j and X_i are disjoint sets. Thus, sets $X_j \cup X_{j_1}$, $X_j \cup X_{j_2}$ and $X_{j_1} \cup X_{j_2}$ are included in sets A_1, A_2 and A_3, respectively, belonging to distinct input partitions.

The following cases must be considered.

Case 1. Assume that A_1 belongs to partition π_3. Since X_{i_1} and X_{j_1} share the subset A, it follows that $A_1 \subseteq Z_3$. Hence A_3 belongs to π_1 or π_2. It is not restrictive to assume that A_3 belongs to π_1. Clearly, it must be that $A_3 \subseteq Z_1$, as $Z_1 \cap A_3 \supseteq A$. Consequently, $X_{j_1} \cup X_{i_1}$ is included in Z_1 and Z_3 thus contradicting the maximality of the 2-components.

Case 2. Assume that A_2 belongs to partition π_3. Being A_2 symmetric to set A_1, this case leads to a contradiction just as in Case 1.

Since Cases 1 and 2 lead to a contradiction, we must assume that A_3 belongs to π_3, with $A_3 \subseteq Z_3$. Clearly, by maximality of 2-components, A_1 and A_2 belong to π_2 and π_1 respectively, where A_1 is distinct from Z_2 and A_2 is distinct from Z_1.

Let X_z be a 2-component different from X_{i_1}, X_{j_1} so that $X_z \cap A \neq \emptyset$. Observe that X_z must belong to π_1 and π_2, otherwise is co-clustered in two partitions with one of X_{i_1}, X_{j_1}, and the maximality of 2-components is violated. Let X_w be a 2-component so that $X_w \cap B \neq \emptyset$. Again observe that X_w must belong to π_1 and π_2, otherwise the maximality of 2-components is violated. But then, since X_z and X_w are in different sets of the same partitions, $X_z \cap X_w = \emptyset$. □

From Lemma 5.4, it follows that if a pair (a,b) of $P_{\pi,\mathrm{OUT},2}(X_i)$ belongs to $P_{\pi,\mathrm{OUT},2}(X_j)$, then there is no other set X_z chosen by the algorithm and such that $(a,b) \in P_{\pi,\mathrm{OUT},2}(X_z)$. Hence Claim 5.5 holds.

Claim 5.5. *A pair (a,b) belongs to at most two sets $P_{\pi,\mathrm{OUT},2}(X_i)$, $P_{\pi,\mathrm{OUT},2}(X_j)$.*

Proof of Lemma 5.1. Let $X_1, \ldots, X_k$ be the sequence of sets added to the solution π by successive iterations of algorithm GREEDY-REDUCED-CC. Define $\mathcal{A}_{X_i} = \{A_j \in \mathcal{A}\colon\ A_j \cap X_i \neq \emptyset\}$. Notice that $\mathcal{A} = \bigcup_{i \leqslant k} \mathcal{A}_{X_i}$ and $P_{\pi,\mathrm{IN}} = \bigcup_{i \leqslant k} P_{\pi,\mathrm{IN}}(X_i)$ and $\bigcup_{i \leqslant k} P_{\pi,\mathrm{OUT}}(X_i) \subseteq P_{\pi,\mathrm{OUT}}$. Now, let us show that $P_{\pi,\mathrm{loss}} = \bigcup_{i \leqslant k} P_{\pi,\mathrm{loss}}(X_i)$. Indeed, observe that $\bigcup_{i \leqslant k} P_{\pi,\mathrm{loss}}(X_i) \subseteq P_{\pi,\mathrm{loss}}$. Thus, let (a,b) be a pair in $P_{\pi,\mathrm{loss}}$. Then (a,b) is in a 2-component C_l of graph G. By construction of the algorithm, there exists a 2-component X_l added at step l by the algorithm GREEDY-REDUCED-CC to π such that $a \in X_l$, while $b \in X_t$, with $l < t$. Consequently, it holds that $(a,b) \in P_{\pi,\mathrm{loss}}(X_l)$, proving that $P_{\pi,\mathrm{loss}} \subseteq \bigcup_{i \leqslant k} P_{\pi,\mathrm{loss}}(X_i)$.

In the following we will prove that:

$$v\big(P_{\pi,\mathrm{IN}}(X_i)\big) + v(\mathcal{A}_{X_i}) + v\big(P_{\pi,\mathrm{OUT},1}(X_i)\big) + \frac{1}{2}v\big(P_{\pi,\mathrm{OUT},2}(X_i)\big) \geqslant 3v\big(P_{\pi,\mathrm{loss}}(X_i)\big).$$

We recall that $v(p) = 2$, for $p \in P_{\pi,\mathrm{OUT},2}(X_i)$, and that, by Claim 5.5, each pair in $P_{\pi,\mathrm{OUT},2}(X_i)$ can be counted at most twice, once for the pairs $P_{\pi,\mathrm{OUT},2}(X_i)$ and once for the pairs $P_{\pi,\mathrm{OUT},2}(X_j)$. Now, by Claim 5.5, $\sum_{i\leqslant k} v(P_{\pi,\mathrm{OUT},1}(X_i)) + \frac{1}{2}v(P_{\pi,\mathrm{OUT},2}(X_i)) \leqslant v(P_{\pi,\mathrm{OUT}}, \pi)$, and thus by applying Claim 5.3 it holds that: $\sum_{i\leqslant k}(v(P_{\pi,\mathrm{IN}}(X_i)) + v(\mathcal{A}_{X_i}) + v(P_{\pi,\mathrm{OUT},1}(X_i)) + \frac{1}{2}v(P_{\pi,\mathrm{OUT},2}(X_i))) \leqslant v(\sum_{i\leqslant k} P_{\pi,\mathrm{IN}}(X_i)) + v(\sum_{i\leqslant k} \mathcal{A}_{X_i}) + v(\sum_{i\leqslant k} P_{\pi,\mathrm{OUT}}(X_i))$. The Lemma 5.1 directly follows.

Lemma 5.6. *Let π be the solution computed by* GREEDY-REDUCED-CC *and X_i the set added to solution π at iteration i, then*

$$v\big(P_{\pi,\mathrm{IN}}(X_i)\big) + v(\mathcal{A}_{X_i}) + v\big(P_{\pi,\mathrm{OUT},1}(X_i)\big) + \frac{1}{2}v\big(P_{\pi,\mathrm{OUT},2}(X_i)\big) \geqslant 3v\big(P_{\pi,\mathrm{loss}}(X_i)\big).$$

Proof. Given $A_r \in \mathcal{A}_{X_i}$, we denote by $P_{\pi,\mathrm{IN}}(X_i, A_r)$ (respectively $P_{\pi,\mathrm{loss}}(X_i, A_r)$) the pairs in $P_{\pi,\mathrm{IN}}(X_i)$ (respectively $P_{\pi,\mathrm{loss}}(X_i)$) having an element in A_r. Denote by $P_{\pi,\mathrm{OUT}}(X_i, A_r)$ the set of pairs in $P_{\pi,\mathrm{OUT}}(X_i)$ having an element in $X_i - A_r$ (such pairs are denoted as $P_{\pi,\mathrm{OUT},1}(X_i, A_r)$) plus the pairs (a, b) in $P_{\pi,\mathrm{OUT},2}(X_i)$ such that a, b belong to X_{r1}, X_{r2} respectively, where X_{r1}, X_{r2} denote the 2-components of graph G_i (G_i is the component graph at iteration i of the algorithm) that share intersection A_r with X_i (such pairs are denoted as $P_{\pi,\mathrm{OUT},2}(X_i, A_r)$). Finally, denote by X_i^{-r} the set $X_i - A_r$ and by X_{r1}^{-r}, X_{r2}^{-r} the set $X_{r1}^{-r} - A_r$ and $X_{r2}^{-r} - A_r$ respectively.

Observe that $P_{\pi,\mathrm{loss}}(X_i, A_r)$ consists of all pairs in $A_r \times X_{r1}^{-r} \cup A_r \times X_{r2}^{-r}$.

In the following we use the following fact: $(*)$ $|X_i^{-r}| \geqslant |X_{r1}^{-r}|$ and $|X_i^{-r}| \geqslant |X_{r2}^{-r}|$. Indeed, by construction X_i is a maximum 2-component w.r.t. the other 2-components of graph G_i built at iteration i of algorithm GREEDY-REDUCED-CC. First we prove Lemma 5.6 when $|X_{r2}^{-r}| = 0$.

Claim 5.7. *Let A_r be a set in $\mathcal{A}$ and assume that $|X_{r2}^{-r}| = 0$.*

$$\frac{1}{2}v\big(P_{\pi,\mathrm{IN}}(X_i, A_r)\big) + v(A_r) + v\big(P_{\pi,\mathrm{OUT},1}(X_i, A_r)\big) \geqslant 3v\big(P_{\pi,\mathrm{loss}}(X_i, A_r)\big). \tag{6}$$

Proof. Observe that $v(P_{\pi,\mathrm{IN}}(X_i, A_r)) = 2|A_r||X_i^{-r}|$, $v(A_r) = \frac{3}{2}|A_r|(|A_r| - 1)$, $v(P_{\pi,\mathrm{OUT},1}(X_i, A_r)) = 2|X_i^{-r}||X_{r1}^{-r}|$ and $v(P_{\pi,\mathrm{loss}}(X_i, A_r)) = |A_r||X_{r1}^{-r}|$.

The set $P_{\pi,\mathrm{IN}}(X_i, A_r)$ consists of pairs in $A_r \times X_i^{-r}$. Let x_r, y be two elements of X_i such that $x_r \in A_r$ and $y \in A_s \subseteq X_i^{-r}$. Observe that each pair (x_r, y) in $P_{\pi,\mathrm{IN}}(X_i, A_r)$ belongs to at most another set $P_{\pi,\mathrm{IN}}(X_i, A_s)$, with $i \neq j$. Hence, we consider $\frac{1}{2}v(P_{\pi,\mathrm{IN}}(X_i, A_r)) = |A_r||X_i^{-r}|$.

The following cases must be considered.

First assume that $|A_r| > 2|X_i^{-r}|$. Then $\frac{3}{2}|A_r|(|A_r| - 1) \geqslant 3|A_r||X_{r1}^{-r}|$.

Assume that $|X_i^{-r}| < |A_r| \leqslant 2|X_i^{-r}|$. Then $\frac{3}{2}|A_r|(|A_r| - 1) \geqslant \frac{3}{2}|A_r||X_{r1}^{-r}|$, $|A_r||X_i^{-r}| \geqslant |A_r||X_{r1}^{-r}|$ and $2|X_i^{-r}||X_{r1}^{-r}| \geqslant |A_r||X_{r1}^{-r}|$ hence the claim holds.

Assume that $|X_i^{-r}| \geqslant |A_r|$. Then $|A_r||X_i^{-r}| > |A_r||X_{r1}^{-r}|$ and $2|X_i^{-r}||X_{r1}^{-r}| \geqslant 2|A_r||X_{r1}^{-r}|$, hence the claim holds. □

Hence in the following we assume w.l.o.g. that $|X_{r1}^{-r}| \geqslant |X_{r2}^{-r}| > 0$. In order to prove Lemma 5.6, we distinguish two cases:

(1) there exists a set A_r in $\mathcal{A}$, $A_r \subseteq X_i$ such that $|A_r| > \frac{3}{2}|X_i^{-r}|$;
(2) there exists no set A_r in $\mathcal{A}$, $A_r \subseteq X_i$ such that $|A_r| > \frac{3}{2}|X_i^{-r}|$;

Case 1

The proof in Case 1 is based on the following bounds.

Claim 5.8. *Assume that X_i verifies Case* 1. *Then for each $A_s \in \mathcal{A}_{X_i}$ such that $A_s \subseteq X_i^{-r}$*

$$v(A_s) + v\big(P_{\pi,\mathrm{OUT},1}(X_i, A_s)\big) \geqslant 3v\big(P_{\pi,\mathrm{loss}}(X_i, A_s)\big). \tag{7}$$

Proof. Let us recall that $P_{\pi,\mathrm{OUT},1}(X_i, A_s)$ consists of pairs in $X_i^{-s} \times X_{s1}^{-s}$ and $X_i^{-s} \times X_{s2}^{-s}$. Since $A_r \subseteq X_i^{-s}$ and $A_s \subseteq X_i^{-r}$, we have $|X_i^{-s}| \geqslant |A_r| > \frac{3}{2}|X_i^{-r}| \geqslant \frac{3}{2}|A_s|$. Consequently $v(P_{\pi,\mathrm{OUT},1}(X_i, A_s)) \geqslant 2|X_i^{-s}|(|X_{s1}^{-s}|+|X_{s2}^{-s}|) > 3|A_s|(|X_{s1}^{-s}| + |X_{s2}^{-s}|) = 3v(P_{\pi,\mathrm{loss}}(X_i, A_s))$. □

Claim 5.9. *Assume that X_i verifies Case 1. Then for $A_r \in \mathcal{A}_{X_i}$*

$$v(A_r) + v\big(P_{\pi,\mathrm{OUT},1}(X_i, A_r)\big) + \frac{1}{2}v\big(P_{\pi,\mathrm{OUT},2}(X_i, A_r)\big) \geqslant 2v\big(P_{\pi,\mathrm{loss}}(X_i, A_r)\big). \tag{8}$$

Proof. Observe that $v(A_r) = \frac{3}{2}|A_r|(|A_r| - 1)$, $v(P_{\pi,\mathrm{OUT},1}(X_i, A_r)) = 2|X_i^{-r}|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$, while $v(P_{\pi,\mathrm{OUT},2}(X_i, A_r)) = |X_{r1}^{-r}||X_{r2}^{-r}|$ and $v(P_{\pi,\mathrm{loss}}(X_i, A_r)) = |A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$. Thus, the following cases must be considered.

Let us first assume that $|X_{r1}^{-r}| + |X_{r2}^{-r}| \geqslant |A_r|$; by fact $(*)$ stated in the above proof of Lemma 5.6, we get $|X_i^{-r}| \geqslant |X_{r1}^{-r}| \geqslant \frac{1}{2}|A_r|$. Thus

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant |A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

while $|X_{r1}^{-r}||X_{r2}^{-r}| \geqslant \frac{1}{2}|A_r||X_{r2}^{-r}|$. Hence, $v(P_{\pi,\mathrm{OUT},1}(X_i, A_r)) + \frac{1}{2}v(P_{\pi,\mathrm{OUT},2}(X_i, A_r)) \geqslant v(P_{\pi,\mathrm{loss}}(X_i, A_r)) + \frac{1}{2}|A_r||X_{r2}^{-r}|$.

Since $|A_r| > \frac{3}{2}|X_i^{-r}|$ and both $|A_r|$ and $|X_i^{-r}|$ are integers, while $\frac{3}{2}|X_i^{-r}|$ can be a fractional number, it follows that $|A_r| - 1 \geqslant \frac{3}{2}|X_i^{-r}| - \frac{1}{2}$. Hence

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) \geqslant \frac{9}{4}|A_r| \cdot \big|X_i^{-r}\big| - \frac{3}{4}|A_r| \geqslant |A_r| \cdot \big|X_{r1}^{-r}\big| + \frac{1}{2}|A_r| \cdot \big|X_{r2}^{-r}\big|.$$

Combining this last inequality with the fact that $v(P_{\pi,\mathrm{OUT},1}(X_i, A_r)) + \frac{1}{2}v(P_{\pi,\mathrm{OUT},2}(X_i, A_r)) \geqslant v(P_{\pi,\mathrm{loss}}(X_i, A_r)) + \frac{1}{2}|A_r||X_{r2}^{-r}|$, the claim follows.

Now, assume that $|X_{r1}^{-r}| + |X_{r2}^{-r}| < |A_r|$ and $|X_{r1}^{-r}| + |X_{r2}^{-r}| \geqslant \frac{1}{2}|A_r|$. We get $\frac{3}{2}|A_r|(|A_r| - 1) \geqslant \frac{3}{2}|A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$ and by fact $(*)$ $|X_i^{-r}| \geqslant |X_{r1}^{-r}| \geqslant \frac{1}{4}|A_r|$. Thus

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{1}{2}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

and the claim easily follows.

Finally, assuming that $|X_{r1}^{-r}| + |X_{r2}^{-r}| < \frac{1}{2}|A_r|$, we get $|A_r| - 1 \geqslant 2(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$ and $\frac{3}{2}|A_r|(|A_r| - 1) \geqslant 3|A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$ and thus the claim holds. □

To conclude the proof, observe that the set $P_{\pi,\mathrm{IN}}(X_i)$ contains the set of the pairs $A_r \times X_i^{-r}$. Since $|X_i^{-r}| \geqslant |X_{r1}^{-r}| \geqslant |X_{r2}^{-r}|$, it follows that $2|A_r||X_i^{-r}| \geqslant |A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$ and thus $v(P_{\pi,\mathrm{IN}}(X_i,)) \geqslant v(P_{\pi,\mathrm{loss}}(X_i, A_r))$ (recall that pairs in $P_{\pi,\mathrm{loss}}$ have value 1, while pairs in $P_{\pi,\mathrm{IN}}$ have value 2). Thus, combining this result with Claims 5.8 and 5.9 yields Lemma 5.6.

Case 2

The proof in Case 2 is based on the following bound.

Claim 5.10. *Assume that X_i verifies Case 2. Then for each $A_r \in \mathcal{A}_{X_i}$,*

$$v(A_r) + v\big(P_{\pi,\mathrm{OUT},1}(X_i, A_r)\big) + \frac{1}{2}v\big(P_{\pi,\mathrm{OUT},2}(X_i, A_r)\big) + \geqslant \frac{5}{2}v\big(P_{\pi,\mathrm{loss}}(X_i, A_r)\big). \tag{9}$$

Proof. Observe that $v(A_r) = \frac{3}{2}|A_r|(|A_r| - 1)$, $v(P_{\pi,\mathrm{OUT},1}(X_i, A_r)) = 2|X_i^{-r}|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$, while $v(P_{\pi,\mathrm{OUT},2}(X_i, A_r)) = 2|X_{r1}^{-r}| \cdot |X_{r2}^{-r}|$, $v(P_{\pi,\mathrm{loss}}(X_i, A_r)) = |A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$. Recall that $|X_{r1}^{-r}| \geqslant |X_{r2}^{-r}|$. Thus, the following cases must be considered.

Assuming $|X_{r1}^{-r}| \geqslant \frac{5}{4}|A_r|$ we get $|X_i^{-r}| \geqslant |X_{r1}^{-r}| \geqslant \frac{5}{4}|A_r|$ and thus

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{5}{2}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

and the claim holds.

Assume now that $|A_r| \leqslant |X_{r1}^{-r}| < \frac{5}{4}|A_r|$, then

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant 2|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big).$$

Since $|A_r| > \frac{4}{5}|X_{r1}^{-r}|$, it follows that $|A_r| - 1 \geqslant \frac{4}{5}|X_{r1}^{-r}| - \frac{4}{5}$. Thus, it holds that:

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) \geqslant \frac{3}{2}|A_r|\left(\frac{4}{5}\big|X_{r1}^{-r}\big| - \frac{4}{5}\right) \geqslant \frac{3}{5}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) - \frac{6}{5}|A_r| \geqslant \frac{1}{2}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) - |A_r|.$$

Finally $|X_{r1}^{-r}||X_{r2}^{-r}| \geqslant |A_r||X_{r2}^{-r}| \geqslant |A_r|$ and the claim holds.

Assume now that $\frac{6}{7}|A_r| \leqslant |X_{r1}^{-r}| < |A_r|$, then $|A_r| - 1 \geqslant |X_{r1}^{-r}| \geqslant |X_{r2}^{-r}|$. Since by hypothesis $|A_r| \leqslant \frac{3}{2}|X_i^{-r}|$, it follows that

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{4}{3}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

and

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) = \frac{7}{6}|A_r|\big(|A_r| - 1\big) + \frac{1}{3}|A_r|\big(|A_r| - 1\big) \geqslant \frac{7}{6}|A_r|\big|X_{r1}^{-r}\big| + \frac{1}{3}|A_r|\big|X_{r2}^{-r}\big|.$$

Moreover, $|X_{r1}^{-r}||X_{r2}^{-r}| \geqslant \frac{6}{7}|A_r||X_{r2}^{-r}|$ and the claim holds.

Assume now that $\frac{2}{3}|A_r| \leqslant |X_{r1}^{-r}| < \frac{6}{7}|A_r|$, then $|A_r| - 1 \geqslant \frac{7}{6}|X_{r1}^{-r}|$. Again, since by hypothesis $|A_r| \leqslant \frac{3}{2}|X_i^{-r}|$, it follows that

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{4}{3}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big).$$

We recall that $|A_r| - 1 \geqslant \frac{7}{6}|X_{r1}^{-r}|$, therefore

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) \geqslant \frac{3}{2}|A_r|\left(\frac{7}{6}\big|X_{r1}^{-r}\big|\right) = \frac{7}{4}|A_r|\big|X_{r1}^{-r}\big| \geqslant \frac{5}{4}|A_r|\big|X_{r1}^{-r}\big| + \frac{1}{2}|A_r|\big|X_{r2}^{-r}\big|.$$

Moreover, $|X_{r1}^{-r}||X_{r2}^{-r}| \geqslant \frac{2}{3}|A_r||X_{r2}^{-r}|$ and the claim holds.

Assuming $\frac{1}{2}|A_r| \leqslant |X_{r1}^{-r}| < \frac{2}{3}|A_r|$ we get $|A_r| - 1 \geqslant \frac{3}{2}|X_{r1}^{-r}|$. Since $|A_r| \leqslant \frac{3}{2}|X_i^{-r}|$, it follows that

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{4}{3}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big).$$

Since $|A_r| - 1 \geqslant \frac{3}{2}|X_{r1}^{-r}|$, it follows that

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) \geqslant \frac{9}{4}|A_r|\big|X_{r1}^{-r}\big| \geqslant \frac{7}{6}|A_r| \cdot \big|X_{r1}^{-r}\big| + \frac{13}{21}|A_r|\big|X_{r2}^{-r}\big|.$$

Moreover, $|X_{r1}^{-r}||X_{r2}^{-r}| \geqslant \frac{1}{2}|A_r||X_{r2}^{-r}|$ and the claim holds.

Assuming $|X_{r1}^{-r}| < \frac{1}{2}|A_r|$, since $|A_r| < \frac{3}{2}|X_i^{-r}|$, we get

$$2\big|X_i^{-r}\big|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big) \geqslant \frac{4}{3}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

and

$$\frac{3}{2}|A_r|\big(|A_r| - 1\big) \geqslant \frac{3}{2}|A_r|\big(\big|X_{r1}^{-r}\big| + \big|X_{r2}^{-r}\big|\big)$$

and the claim holds. □

Now, the set $P_{\pi,\mathrm{IN}}(X_i, A_r)$ consists of pairs in $A_r \times X_i^{-r}$. Since $|X_i^{-r}| \geqslant |X_{r1}^{-r}| \geqslant |X_{r2}^{-r}|$, it holds that $2|A_r|(|X_i^{-r}|) \geqslant |A_r|(|X_{r1}^{-r}| + |X_{r2}^{-r}|)$, that is $v(P_{\pi,\mathrm{IN}}(X_i, A_r)) \geqslant v(P_{\pi,\mathrm{loss}}(X_i, A_r))$. Let x_r, y be two elements of

X_i such that $x_r \in A_r$ and $y \in A_s \subseteq X_i$. Since each pair (x_r, y) in $P_{\pi,\mathrm{IN}}(X_i)$ belongs to at most two sets of pairs $P_{\pi,\mathrm{IN}}(X_i, A_r)$, $P_{\pi,\mathrm{IN}}(X_i, A_s)$, it follows that $\frac{1}{2}v(P_{\pi,\mathrm{IN}}(X_i)) \geqslant \frac{1}{2}v(P_{\pi,\mathrm{loss}}(X_i))$. Thus, combining this result with Claim 5.10 gives Lemma 5.6. □

Acknowledgments

We would like to express our deep appreciation to the anonymous reviewers for their thorough reviews of the manuscript and many very helpful and constructive comments. TJ's research is supported in part by NSF grant CCR-0309902, National Key Project for Basic Research (973) grant 2002CB512801, NSFC grant 60528001, and a Changjiang Visiting Professorship at Tsinghua University. PB's research and RD's research is supported by FAR 2006—Università degli Studi di Milano-Bicocca grant "Algorithms for System Biology and Bioinformatics" and MIUR grant FIRB-LIBI "Laboratorio Internazionable di Bioinformatica." GDV's research is supported in part by FAR 2006—Università degli Studi di Milano-Bicocca grant "Algoritmi efficienti per la ricostruzione e il confronto di storie evolutive."

References

[1] N. Ailon, M. Charikar, A. Newman, Aggregating inconsistent information: Ranking and clustering, in: Proceedings of the 37th Annual Symposium on Theory of Computing, STOC, 2005, pp. 684–693.
[2] P. Alimonti, V. Kann, Some APX-completeness results for cubic graphs, Theoret. Comput. Sci. 237 (1–2) (2000) 123–134.
[3] S. Arora, D. Karger, M. Karpinski, Polynomial time approximation schemes for dense instances of $\mathcal{NP}$-hard problems, J. Comput. System Sci. 58 (2000) 193–210.
[4] S. Arora, A. Frieze, H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arrangement problems, Math. Program. 92 (1) (2002) 1–36.
[5] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, Springer-Verlag, 1999.
[6] N. Bansal, A. Blum, S. Chawla, Correlation Clustering, in: Proc. 43rd Symp. Foundations of Computer Science, FOCS, 2002, pp. 238–247.
[7] M. Charikar, V. Guruswami, A. Wirth, Clustering with qualitative information, in: Proc. 44th Symp. Foundations of Computer Science, FOCS, 2003, pp. 524–533.
[8] E.D. Demaine, N. Immorlica, Correlation Clustering with partial information, in: Proc. 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX, 2003, pp. 1–13.
[9] D. Emanuel, A. Fiat, Correlation Clustering—Minimizing disagreements on arbitrary weighted graphs, in: Proc. 11th European Symp. on Algorithms, ESA, 2003, pp. 208–220.
[10] V. Filkov, S. Skiena, Integrating microarray data by Consensus Clustering, in: Proc. 15th International Conference on Tools with Artificial Intelligence, ICTAI, 2003, pp. 418–425.
[11] V. Filkov, S. Skiena, Heterogeneous data integration with the Consensus Clustering formalism, in: Data Integration in the Life Sciences, First International Workshop, DILS, 2004, pp. 110–123.
[12] T. Jiang, P. Kearney, M. Li, A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application, SIAM J. Comput. 30 (6) (2000) 1942–1961.
[13] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, in: Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 2005, pp. 341–352.
[14] M. Grötschel, Y. Wakabayashi, A cutting plane algorithm for a clustering problem, Math. Program. 45 (1989) 52–96.
[15] D. Gusfield, Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press, Cambridge, 1997.
[16] D.S. Hochbaum, D.B. Shmoys, A unified approach to approximate algorithms for bottleneck problems, J. ACM 33 (3) (1986) 533–550.
[17] M. Krivanek, J. Moravek, Hard problems in hierarchical-tree clustering, Acta Inform. 23 (1986) 311–323.
[18] C. Swamy, Correlation clustering: Maximizing agreements via semidefinite programming, in: Proc. 15th Symp. on Discrete Algorithms, SODA, 2004, pp. 526–527.
[19] Y. Wakabayashi, The complexity of computing medians of relations, Resenhas 3 (3) (1998) 323–349.